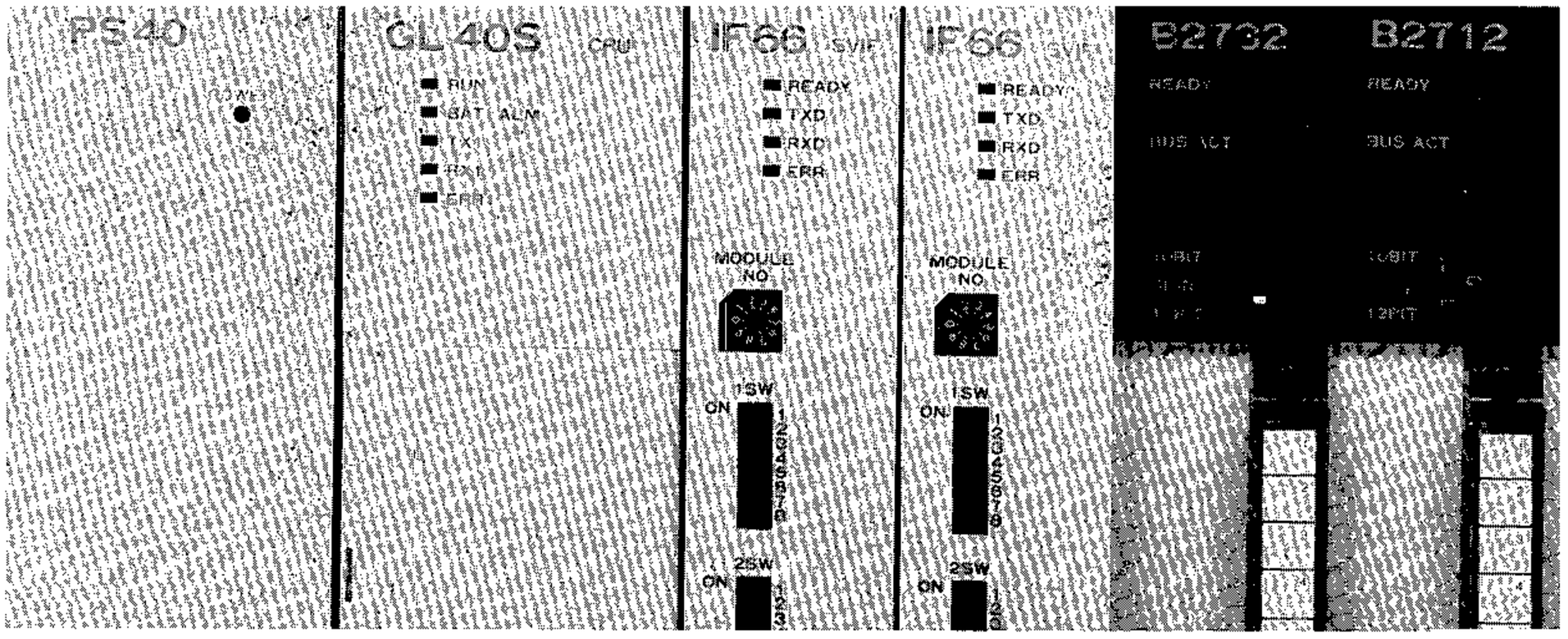
YASKAWA

MEMOCON-SC GL40S
DESCRIPTIVE INFORMATION

PROGRAMMABLE CONTROLLER
DESIGN AND MAINTENANCE



YASKAWA

MANUAL NO. SIE-C815-15.1C

NOTES FOR SAFE OPERATION

Read these manuals thoroughly before use of MEMOCON-SC GL40S. In these manuals, NOTES FOR SAFE OPERATION are classified as "WARNING" and "CAUTION."

⚠WARNING : Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.

⚠ CAUTION : Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.

Even items described in ⚠ CAUTION may result in a vital accident in some situations. In either case, follow these important notes.

The following shows the symbols of prohibition and mandatory action.

⃠ PROHIBITION : Specifies prohibited handling.

MANDATORY ACTION : Specifies actions that must be taken.

After reading these manuals, keep them readily available for those using the equipment.

1 INSTALLATION

⚠ CAUTION

- The installation environment must meet the environmental conditions given in the product catalog and manuals.
  Using the GL40S in environments subject to high temperatures, high humidity, excessive dust, corrosive gases, vibration, or shock can lead to electric shock, fire, or faulty operation.

  Do not use the GL40S in the following locations.
  - Locations subject to direct sunlight or ambient temperatures not between 0 and 55℃.
  - Locations subject to relative humidity in excess of 95%, rapid changes in humidity, or condensation.
  - Locations subject to corrosive or flammable gas.
  - Locations that would subject the GL40S to direct vibration or shock.
  - Locations subject to contact with water, oil, chemicals, etc.

- Install products correctly according to the instructions.
  Improper installation may result in accidents or malfunctions.
  ① Be sure all screws are tight.
  All screws for installation and terminal board should be securely tightened and checked for loosening. Malfunctions in the GL40S may occur as a result of loose screws.
  ② Install the mounting base correctly.
  Install the mounting base facing in the correct direction. Incorrect installation may result in accidents or malfunctions.

- When installing the mounting base, leave the cover on to prevent contamination from foreign matter.
  Foreign matter can cause malfunction in the GL40S.

- Do not remove the cover of the connector where a module is not mounted.
  Foreign matter can cause malfunction.

2 WIRING

⚠ CAUTION

- Connect a power supply complying with the rated specifications.
  A power supply that does not comply with the rating may cause a fire.
- Wiring must be performed by qualified personnel.
  Mistakes in wiring can cause fires, product failure, or malfunctions.
- When wiring, do not allow foreign matters such as wire ship to enter the mounting base or the module.
  Foreign matter can cause fires, product failures, or malfunctions.
- When using output modules without built-in fuses, connect the fuse in series with the load to conform to load specifications.
  Unconnected fuses may cause fire, breakdown, and damage output circuits.

⏚ MANDATORY ACTION

- Ground the protective ground terminal to a resistance of 100 Ω max.
  Failure to observe this instruction may result in electric shock or malfunction.

INSERT THE INTERFACE CABLES PROPERLY

- Insert the connectors of the various interface cables that are connected to GL40S into the communication parts and secure them properly.
  Failure to observe this instruction may result in malfunctions.

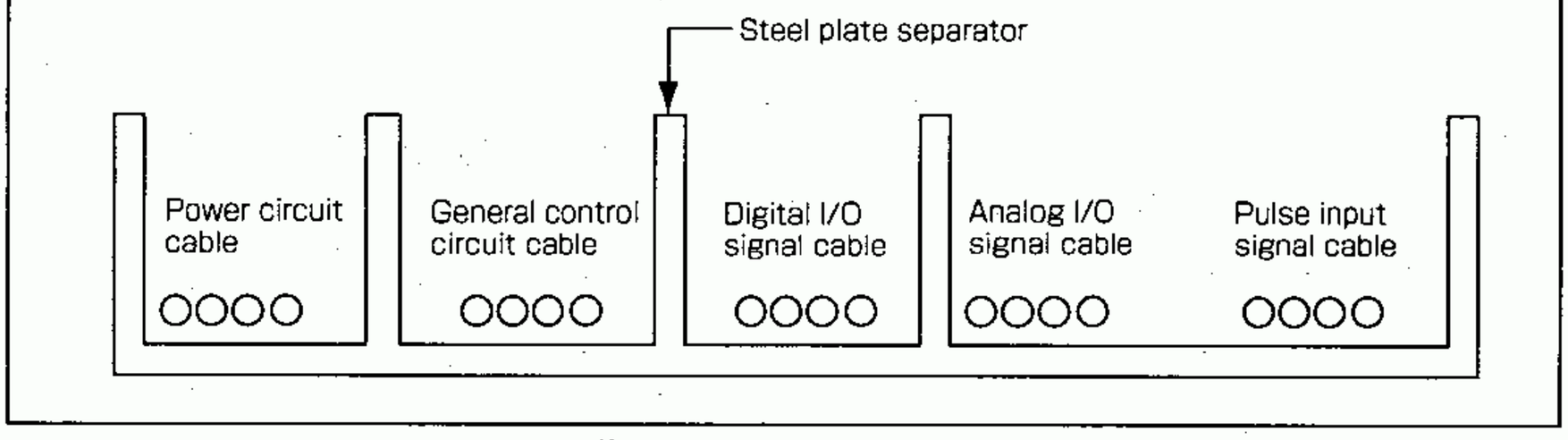
NOISE REDUCTION MEASURES

- When noise from external power supply lines causes problems, install an insulated transformer and noise filter for effective noise prevention.
  Insufficient noise reduction measure may cause malfunctions in the GL40S.

⚠ SEPARATE WIRING PROPERLY

- I/O lines connecting external devices to the GL40S must be selected based on the following considerations:
  mechanical strength, resistance to noise, wiring distance, signal voltage, etc.
- I/O lines must be separated from power lines both within and outside of the control panel to minimize the affects of noise. Faulty operation can result if I/O lines are not sufficiently separated from power lines.

(Example of external wiring)



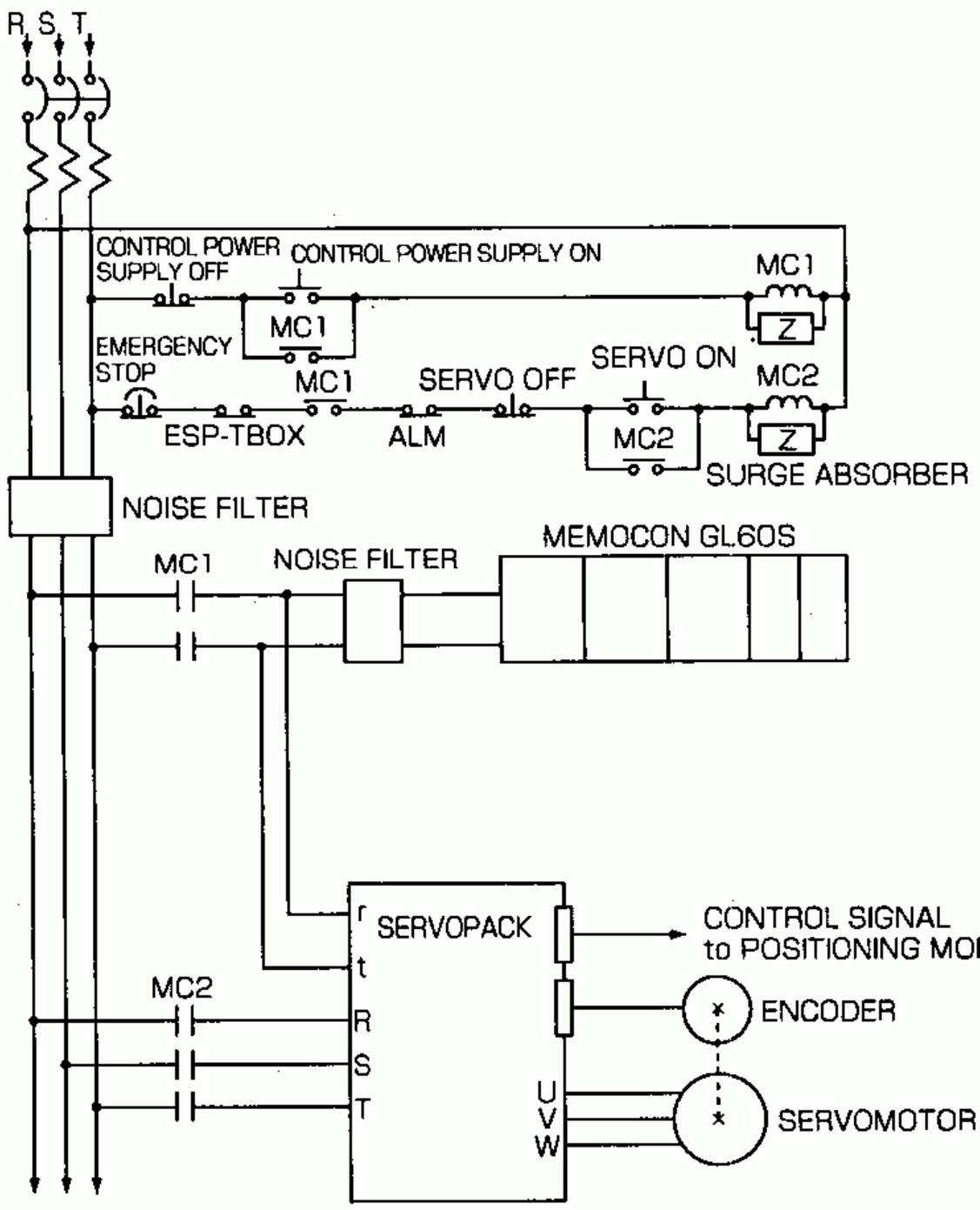
3 PRECAUTION UPON USE

⚠ WARNING

- Do not touch module terminals under current conditions.
  There is danger of electric shock.
- Provide an emergency stop circuit, interlock circuit, etc. at the exterior of GL40S.
  Failure to observe this instruction may result in injury or damage to equipment.

Provide an emergency stop circuit at the exterior of GL40S.

Do not install the emergency stop circuit as part of the MEMOCON-based ladder program. Install an emergency stop circuit to an external relay as shown below.
Use a NC contact (mechanical contact) to connect the emergency stop switch. The emergency stop switch should cut off the main power supply when depressed.
If these steps are not followed, the emergency switch will not engage even if input circuits are damaged or cables are cut. Failure to follow instructions may cause damage to machines and injury to personnel.

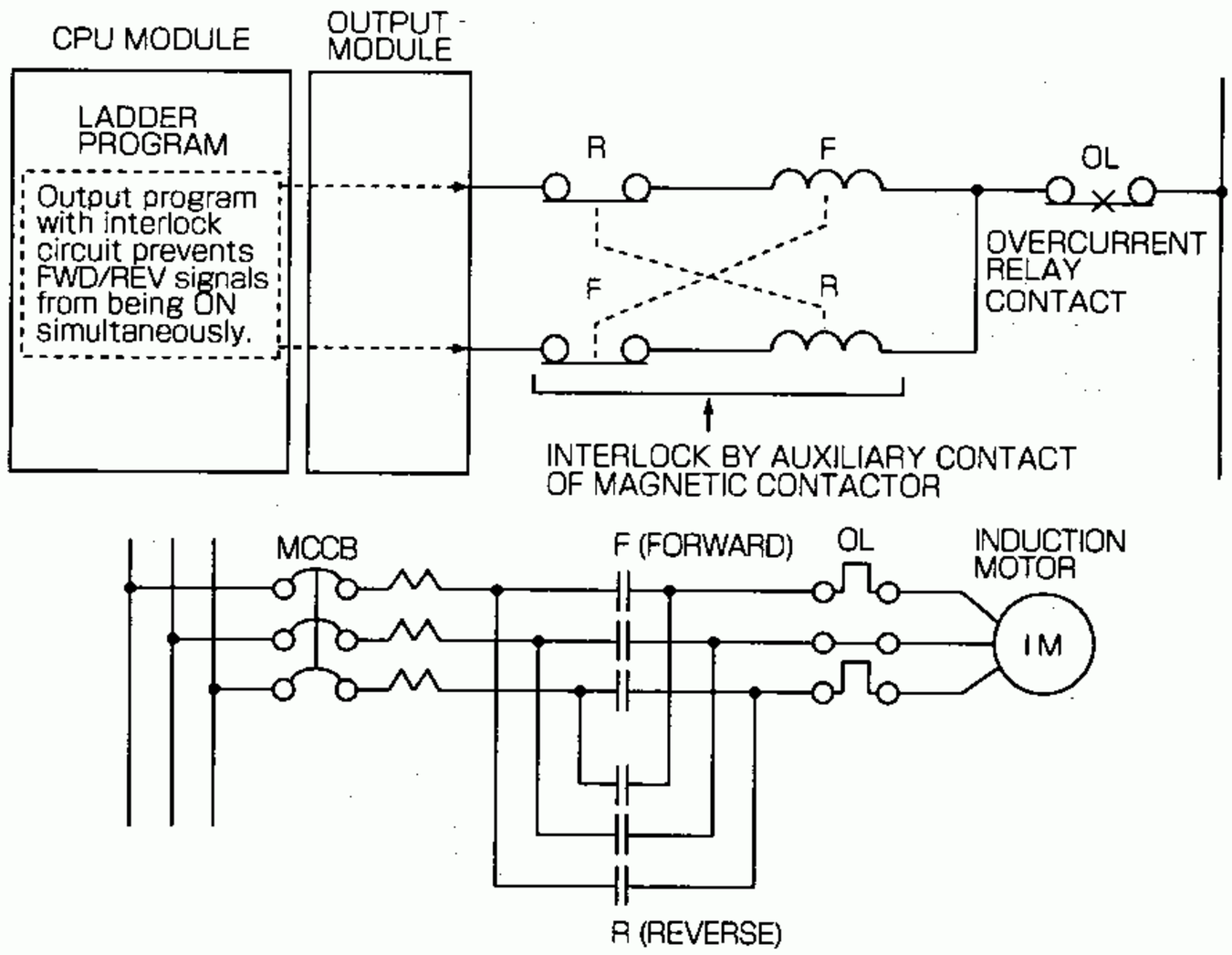


(Continued)

Provide an interlock circuit at the exterior of the GL40S.

Provide an interlock circuit at the exterior of the GL40S to prevent injury or damage to equipment.

(Example) Interlock phase for forward and reverse drives
To prevent forward and reverse drive signals from being turned on simultaneously, install an interlock circuit in the GL40S ladder program.
At the same time, use auxiliary contacts of external magnetic contactor to install a second interlock circuit to prevent forward and reverse drive magnetic contactors from being turned on simultaneously.



⚠ CAUTION

- When using output modules without built-in fuses, connect the fuse in series with the load to conform to load specifications.
  Unconnected fuses may cause fires, breakdowns, or damage output circuits.

4 MAINTENANCE

⚠ WARNING

- Do not reverse polarity, charge, disassemble, or expose battery to heat or flame. There is danger of bursting or fire.

PROHIBITION

- Do not attempt to disassemble or modify the MEMOCON-SC in any way. Doing so can cause fires, product failure, or malfunctions.

MONITOR THE LIFE OF BATTERY

- Monitor the life of CPU module built-in battery.
  If the "BATTERY ALARM" indicator lights, replace the battery with a new one within a month. CPU module memory (ladder program, etc.) may be erased if battery change is delayed.

OVERHAUL POWER MODULE REGULARLY

- Perform power module overhaul every 5 years.
  Malfunction in the power supply unit may occur due to deterioration in product lifetime of smoothing condensor, etc. Shortening overhaul frequency should be considered under the following conditions:
  - When used in areas of high humidity or heavy climate change.
  - When there are large fluctuations in electrical voltage, load, frequency and waveform.
  - When equipment was stored in a harsh environment or left unused for long periods.

5 GENERAL PRECAUTION

- GL40S was not designed or manufactured for use in devices or systems that concern peoples' lives.
  Users who intend to use the product described in this manual for special purposes such as devices or systems relating to transportation, medical, space aviation, atomic power control, or underwater use must contact YASKAWA representatives beforehand.
- This product has been manufactured under strict quality control guidelines. However, if this product is to be installed in any location in which a failure of GL40S involves a life and death situation or in a facility where failure may cause a serious accident, safety devices must be installed to minimize the likelihood of any accident.
- Any illustrations, photographs, or example used in this manual are provided as examples only and may not apply to all product to which this manual is applicable.
- The products and specifications described in this manual or the content and presentation of the manual may be changed without notice to improve the product and/or the manual.
  A new version of the manual will be re-released under a revised document number when any changes are made.
- Contact your YASKAWA representative listed on the back of this manual to order a new manual whenever this manual is damaged or lost.
  Please provide the document number listed on the front cover of this manual when ordering.
- Contact your YASKAWA representative listed on the back of this manual to order new nameplates whenever a nameplate becomes worn or damaged.
- YASKAWA cannot make any guarantee for products which have been modified.
  YASKAWA assumes no responsibility for any injury or damage caused by a modified product.

OVERVIEW OF MANUAL

- This manual describes the following items of GL40S.
  ① System configuration
  ② System function and specification
  ③ Operation function
  ④ SFC function
  ⑤ Installation and wiring
  ⑥ Internal board installation and mounting hole dimensions
  ⑦ Dimensions
- Read this manual carefully in order to use the GL40S properly. Also, keep this manual in a safe place so that it can be used whenever necessary.
- Refer to the following manuals as necessary.

	Document Title	Document Number	Content
I/O Modules	MEMOCON-SC GL40S, GL60S, GL70H 2000 Series I/O Modules DESCRIPTIVE INFORMATION	SIE-C815-13.3	Describes functions, specifications, application methods, etc., for the 2000 Series Digital I/O Modules.
	MEMOCON-SC GL40S, GL60S, GL70H 2000 Series Analog I/O Modules DESCRIPTIVE INFORMATION	SIE-C815-13.9	Describes functions, specifications, application methods, etc., for the 2000 Series Analog I/O Modules.
Intelligent Modules	MEMOCON-SC GL40S, GL60S, GL70H 2000 Series Reversible Counter Module DESCRIPTIVE INFORMATION	SIE-C815-13.11	Describes functions, specifications, application methods, etc., for the 2000 Series Reversible Counter Module (B2801).
	MEMOCON-SC GL40S, GL60S, GL70H 2000 SERIES POSITIONING MODULE B2803 DESCRIPTIVE INFORMATION	SIE-C815-13.13	Describes functions, specifications, application methods, etc., for the 2000 Series Positioning Module B2803.
	MEMOCON-SC GL40S, GL60S, GL70H 2000 SERIES I/O POSITIONING MODULE B2833 DESCRIPTIVE INFORMATION	SIE-C815-13.17	Describes functions, specifications, application methods, etc., for the 2000 Series Positioning Module B2833.
Communi-cation Modules	MEMOCON-SC GL40S, GL60S, GL70H MEMOBUS DESCRIPTIVE INFORMATION	SIE-C815-13.60	Describes functions, specifications, application methods, etc., for MEMOBUS.
	MEMOCON-SC GL40S, GL60S, GL70H 2000 Series PC Link Module DESCRIPTIVE INFORMATION	SIE-C815-14.8	Describes functions, specifications, application methods, etc., for the 2000 Series PC Link Module (IF64).
	MEMOCON-SC GL40S, GL60S, GL70H 2000 Series Uni-Wire Interface Module DESCRIPTIVE INFORMATION	SIE-C815-14.32	Describes functions, specifications, application methods, etc., for the 2000 Series Uni-wire Interface Module (B2808).

	Document Title	Document Number	Content
AC Servo Drive	AC Servo Drives HR Series DESCRIPTIVE MANUAL For Multi-functions/Positioning control	TSE-S800-6	Describes functions, specifications, application methods, etc., for the AC Servo Drive HR Series.

* Thoroughly check the specifications and conditions or restrictions of the product before use.

USING THIS MANUAL

- This manual concerns those people involved with the following activities.
  ① Preparing an estimate for the GL40S
  ② Evaluating the GL40S for use
  ③ Design and setup of GL40S installed control panels and operation panels
  ④ Manufacture of GL40S installed control panels and operation panels
  ⑤ Inspection of GL40S installed control panels and operation panels
  ⑥ Test run adjustment of GL40S installed control panels and operation panels
  ⑦ Maintenance of GL40S installed control panels and operation panels

- Meaning of Basic Terms
  In this manual, the following terms indicate the meanings as described below, unless otherwise specified.
  - PC = Programmable Controller
  - PP = Programming Panel
  - GL40S, GL60S = MEMOCON-SC GL40S, GL60S, GL60H, GL70H Programmable
    GL60H, GL70H Controllers

CONTENTS

CONTENTS

INDEX

INDEX (Cont'd)

SECTION 1
INTRODUCTION

Memocon-SC GL40S (hereafter called GL40S) is a middle-scale programmable controller (hereafter called PC) that meets the requirements of the CIM age with enhanced functions of arithmetic operations, data processing, and communications.

GL40S has greatly improved motion control to enable positioning control that is possible only with high-grade PCs.

[Features]

1. Numerical Operation and Data Processing Functions.

- Data length is 16 bit.
- 2048 data-storing registers are provided.
- Double-length arithmetic operation is possible.
- Signed arithmetic operation is possible.

2. Motion Control

- Flexible multi-axis motion control is achievable with the JAMSC-IF66 servo interface and software servo.
- Specially provided motion control commands realize easy programming and easy-to-see programs.
- Motion control by 2000-series positioning modules is possible.

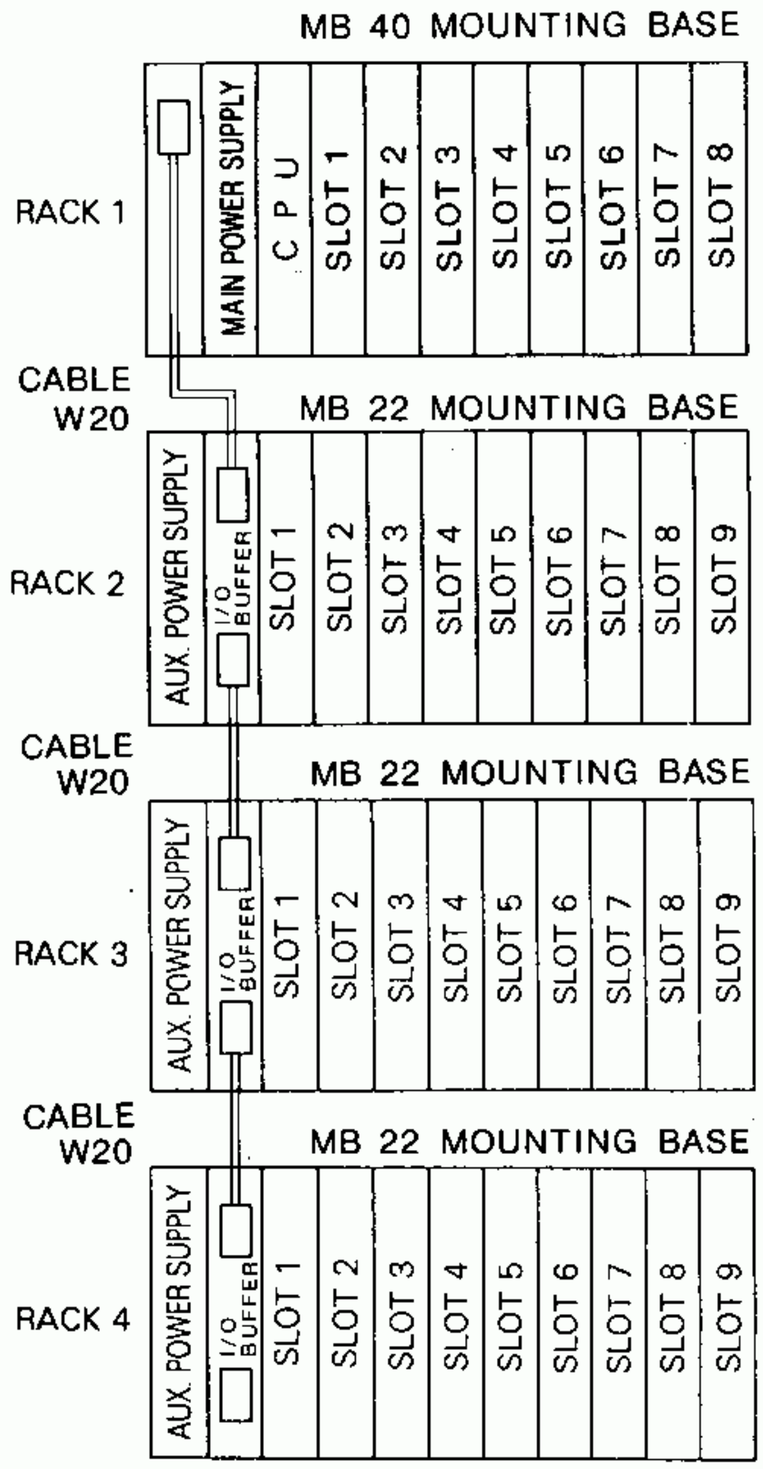
3. Communication Functions

- MEMOBUS communication by the master PC, FA monitor ACGC, or host computer is possible.
- High-speed mutual communication among PCs is possible by the JAMSC-IF66 PC Link module. Communication with the GL60S and CP series is possible with the PC Link.
- Communication with all PCs connected to PC Link is possible by use of a programming panel.
- Using communication commands, GL40S performs communication as the master.

SECTION 2
GL40S CONFIGURATION

Table 2.1 GL40S Configuration

Component	Description
CPU Module (CPU) GL40S1, S2, S3 2 k 4 k 8 k	The CPU module includes a logic solver and memory. The ladder circuits are stored in the memory and solved according to input data sent from an I/O driver. The results are output to the I/O driver. The program memory is available in 2 k /4 k /8 k words. The GL40S is capable of dealing with discrete inputs/outputs (ON/OFF signals) of up to 512 points and up to 128 register inputs/outputs (5-digit decimal or 16-bit binary data). This module includes one RS-232C port (MEMOBUS) for communication with the Programming Panel and a computer.
Main Power Supply Module PS40	The main power supply module supplies DC power to CPU module, optional modules and I/O modules in rack 1 of CH1.
Auxiliary Power Supply Module PS21	The auxiliary power supply module supplies DC power to I/O buffer module and I/O modules in each expanding I/O rack.
Expanding Communication Module (COMM) IF61	This module is used to expand the communication ports. For communication with the programming panel and a computer, two RS-232C ports (MEMOBUS) are also available.
Expanding Communication Module (COMM) (with RAP port) IF41A	The I/O processor module includes two RS-232C ports (MEMOBUS) for communication with the Programming Panel and a computer. By operating the front Register Access Panel (RAP), it is possible to display the status of the coils and input relays, to perform simulation (forced ON/OFF), to display and alter the contents of the registers, and to set and display communication parameters. I/O driver is incorporated communication module.
PC Link Module (LINK) IF64	This module performs high-speed data transmitting and receiving among the CPUs of GL40S and GL60S PCs and has one MEMOBUS port, and the function of programming and monitoring CPUs up to 32 units without a modem, connected to the programming panel.
Servo Interface Module (SVIF) IF66	This module can drive 1 to 8 software servo-controlled motors with an absolute encoder, and supports point-to-point independent multi-axis positioning or 2- or 3-axis linear interpolation positioning. Since up to 4 servo interface modules can be connected to a CPU, a maximum of 32 (4 × 8) motors can be controlled.
I/O Buffer Module (IOB) B2110A	The I/O buffer module is used for rack 2 or higher.
RAP (Register Access Panel) Module IF69	• For coil or input relay: Monitoring ON/OFF status, disabling (forced ON/OFF). • Displaying or altering register contents. • Setting or displaying communication parameters. • Monitoring ON/OFF status of coil or input relay by status indication LED. • 1-scan pulse monitoring available.
I/O Modules (2000 Series)	• Discrete signal modules: One module is provided with inputs or outputs of 16, 32 or 64 circuits. It is usable for numeric signals (by I/O allocation). • Numeric signal modules: One module is provided with eight numeric inputs or outputs of 16 bits. • Analog modules: An A/D converter module has eight circuits and a D/A converter module of two circuits. • Other modules: Counter module, positioning module
Mounting Base MB40, MB22	The CPU module, power supply module, peripheral modules, and I/O modules are mounted on a mounting base. The type of the mounting base (two types) varies with the type of module. The modules mounted on the base are connected to each other via a built-in mother board. Connections between mounting bases are made with I/O cables.
Programming Panel P150, P140	The programming panel permits storing a program, altering or deleting the stored program, monitoring status, and printing out a ladder diagram through a connected printer.



Note

1. Mounting bases are used up to 4 stairs. Various small bases can also be used.

2. In the I/O part, I/O modules corresponding to the maximum of 35 slots can be mounted, where the combination of inputs and outputs is free. However, the following conditions place restrictions.

   - Discrete inputs + discrete outputs ≤ 512
   - Register inputs + register outputs ≦ 128

3. Option modules such as the COMM module (Extended communication module), PC Link module, and servo interface modules must be mounted to the MB40 mounting base.

Fig. 2.1 GL40S Configuration

SECTION 3
GL40S SPECIFICATIONS

3.1 BASIC GL40S SPECIFICATIONS

Table 3.1 Basic GL40S Specifications

Items	Specifications
Power Supply	Single-phase 85 to 132 (121) VAC, 47.5 to 63Hz
Consumed Power	100VA (main power supply module), 70 VA (aux. power supply module)
Holding Time	10 ms (Less than 10 ms is not regarded as a power failure.)
Ambient Temperature	0 to + 55℃ (excluding peripheral devices)
Storage Temperature	−20℃ to + 85℃ (excluding lithium battery)
Humidity	30% to 95% relative (non-condensing)
Vibration-Resistance	In compliance with JIS* C 0911 (excluding peripheral devices)
Shock-Resistance	10 G max (excluding peripheral devices)
Environmental Condition	Free from explosive, inflammable, corrosive gases
Grounding	Grounding resistance: 100Ω or less
Dielectric Strength	1500 VAC for 1 minute
Insulation Resistance	100 MΩ or more at 500 VDC
Noise Immunity	1500 Vp-p, pulse width: 1 μs, rising time: 1 ns

*Japanese Industrial Standard

3.2 MODULE SPECIFICATIONS

3.2.1 CPU Module

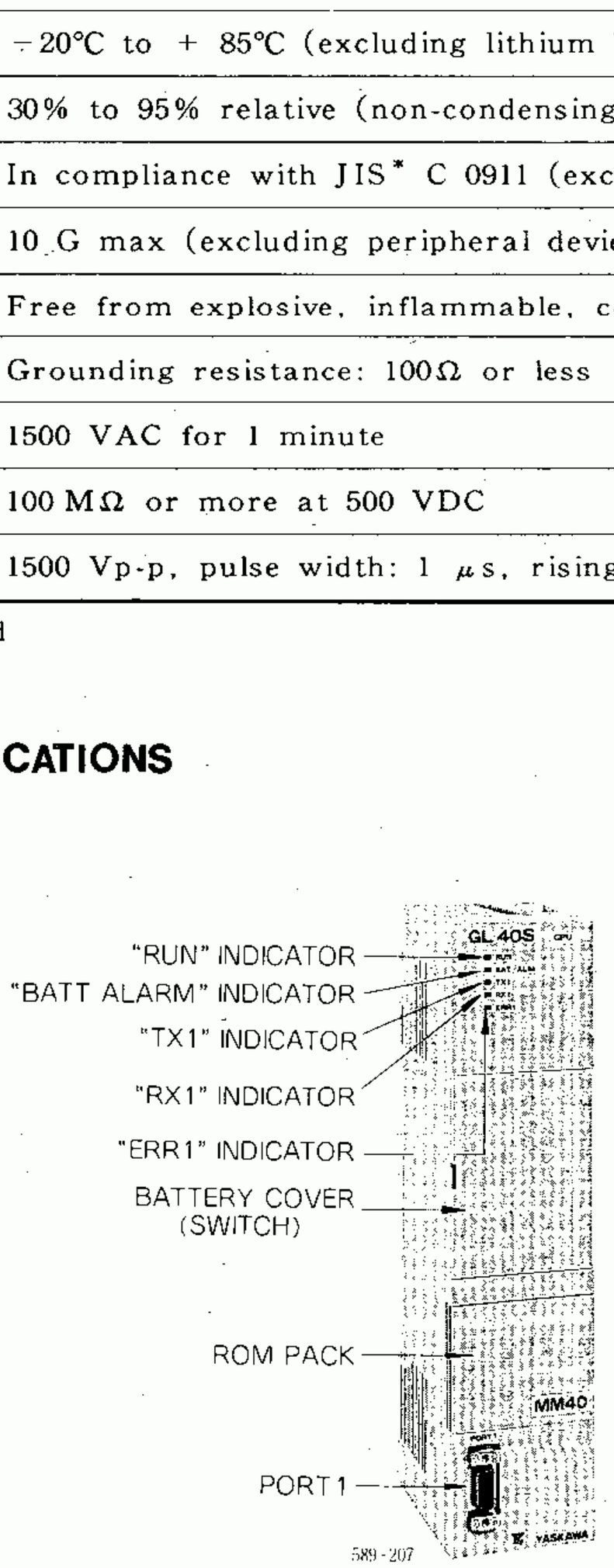


Fig. 3.1 CPU Module

Table 3.2 CPU Module Specifications

Items	Specifications	
Type	DDSCR-GL40S1, GL40S2, GL40S3	
Control Method	Stored program and scan control	
Programming	Relay ladder diagram symbology	
Program Memory Size	2k/4k/8k words: 24-bit per word (Standard: CMOS RAM with battery back-up, Option: EPROM, EEPROM)	
Data Memory Size	2048 words holding registers, 16-bit per word	
Scan Time	0.125 μs per word (basic instruction)	
Logic Function	Relay	• Normally open contact, normally closed contact • Transitional contact (OFF to ON), or (ON to OFF) • Horizontal shunt, vertical shunt, vertical open • Coil, latched coil
	Timer	• Type: Seconds, tenths of seconds, hundredths of seconds • Maximum preset value: 4-digit decimal • Setting available from external device
	Counter	• Up counter, down counter • Maximum preset value: 4-digit decimal • Setting available from external device
	Arithmetic	• (Double-precision) addition, (double-precision) subtraction, (double-precision) multiply, (double-precision) divide (in 4- or 8-digit decimal)
	Arithmetic with Sign	(Double-precision) addition with sign, (double-precision) subtraction with sign, multiply with sign, divide with sign (in 4- or 8-digit decimal)
	Square Arithmetic	Square root (SQRT), Double square root (DSQR)
	Trigonometric Function	Sine, Cosine
	Move	R → T, T → R, T → T, BLKM, FIN, FOUT, SRCH, STAT, TEST
	Move with Index	• DIBT, DIBR • SIBT, SIBR
	Data Convert	BIN, BCD, SWAP, SORT, BYSL, BYCM, BADD
	Matrix	AND, OR, XOR, COMP, CMPR, MBIT, SENS, BROT, MROT, TWST, BCNT
	Special Function	GOSUB, SKIP, STEPPING RELAY, 15 MOTION COMMANDS
	Communication	COMM
Input/Output Points	• Local I/O discrete: Input + Output ≦ 512 points register: Input + Output ≦ 128 words	
PP Port (Port 1)	RS-232C 1 port transmission parameter fixed: RTU, 9600 bps, 1 stop bit, even parity	
Indicating Lamp	• RUN: Lights when CPU module is proper in operation. • BATT ALARM: Lights when the output voltage of CMOS RAM back-up battery is low level, with AC power supply turned on. • TX1: Lights at PP port transmitting. (bit stream of data) • RX1: Lights at PP port receiving. (bit stream of data) • ERR1: Lights at PP port communication error. (Lights more than 50 ms)	
Switch	Address switch	Device address of MEMOBUS (1 to 99: decimal) ····· rotary switch × 2
	Memory protect	Toggle switch
Diagnostic Function	• Checksum of memory • Watchdog timer checking • Battery monitoring • Internal code checking • Reference number checking • I/O allocation checking • Memory diagnostic	
Backed-up Memory (For only CMOS)	• Type: 1-lithium battery • Battery life: 5 years, at 25 ℃ • Memory contents holding time: 1 year, at 25 ℃	
Mounting Location	On mounting base MB40 (CPU base)	
Dimensions in mm	60 (W) × 250 (H) × 100 (D)	
Approx Weight	0.7 kg	

3.2.2 ROM Pack

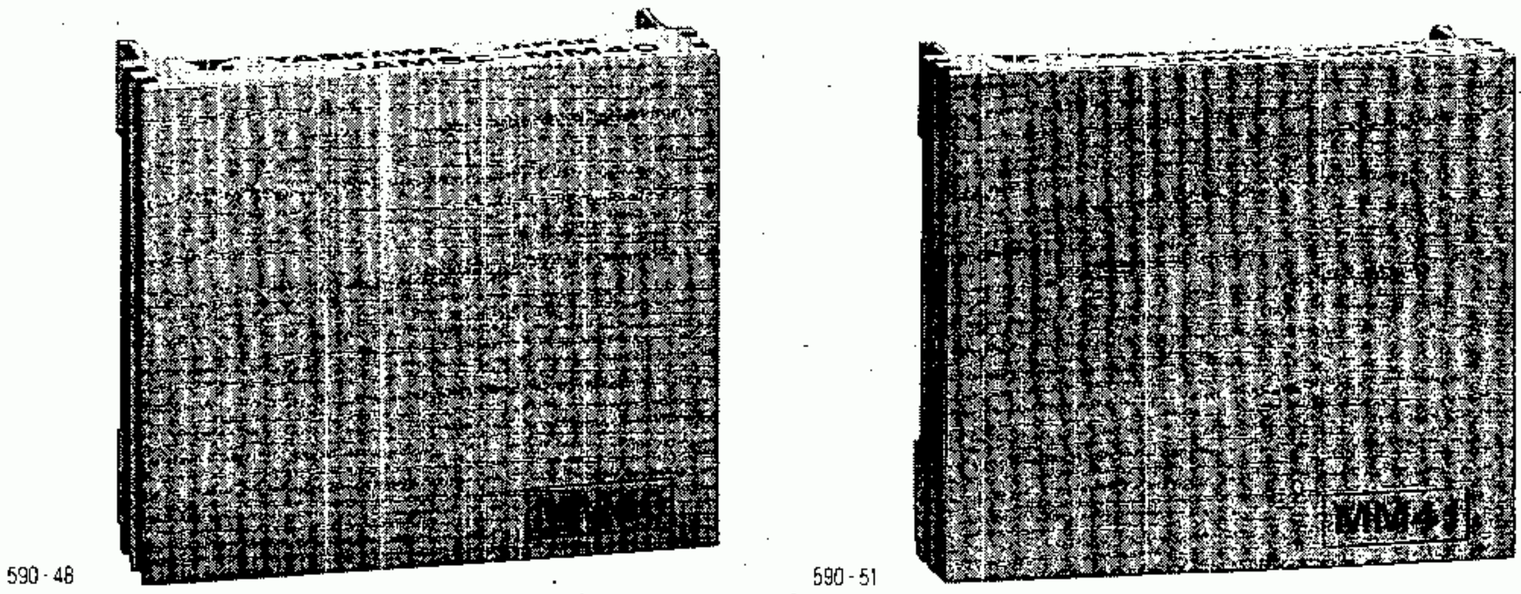


Fig. 3.2 ROM Pack Module

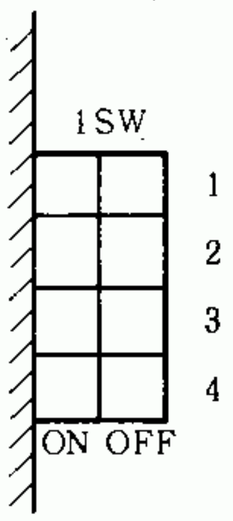
Table 3.3 ROM Pack

ROM Pack Type	ROM Type	Access Time	Remarks
JAMSC-MM40	MBM27C 256	250 ns	EPROM
JAMSC-MM41	X28C256	250 ns	EEPROM

Table 3.4 CPU Built-in Dip Switch Setting

1SW -1	1SW -2	1SW -3	User Program	I/O Allocation	Holding Register	Latch Coil	Disable Coil	History of Coil, Input Relay	Memory Type
OFF	×	×	RAM	RAM	RAM	RAM	RAM	Update	RAM
ON	OFF	OFF	ROM→RAM	ROM→RAM	Clear to 0	OFF	Clear	Clear	EPROM
ON	OFF	ON	ROM→RAM	ROM→RAM	ROM→RAM	OFF	Clear	Clear	
ON	ON	OFF	ROM→RAM	ROM→RAM	Clear to 0	OFF	Clear	Clear	EEPROM
ON	ON	ON	ROM→RAM	ROM→RAM	ROM→RAM	OFF	Clear	Clear	

1SW -1	1SW -4	Operation
ON	OFF	After transmission ROM → RAM, CPU starts
ON	ON	After transmission ROM → RAM, CPU stops



Built-in Dip Switch

3.2.3 Power Supply Module

(1) Main Power Supply Module

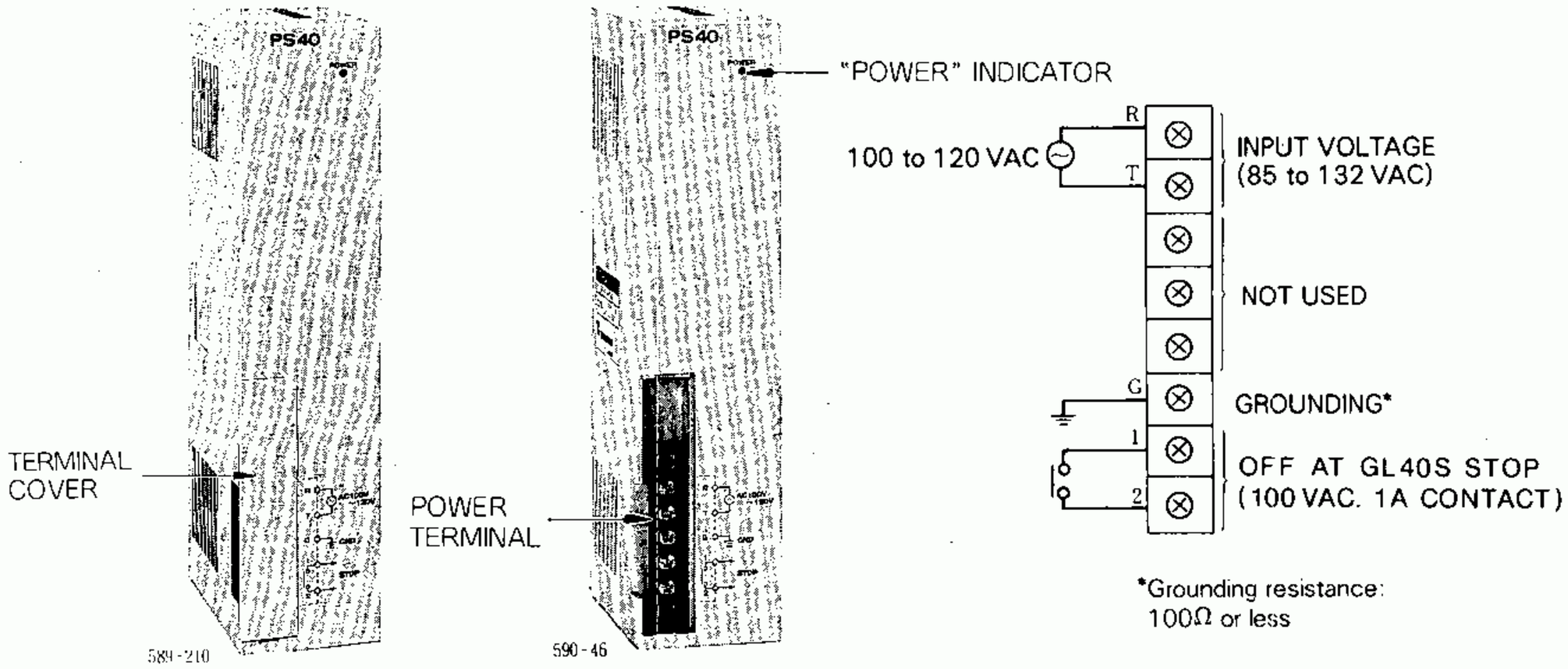
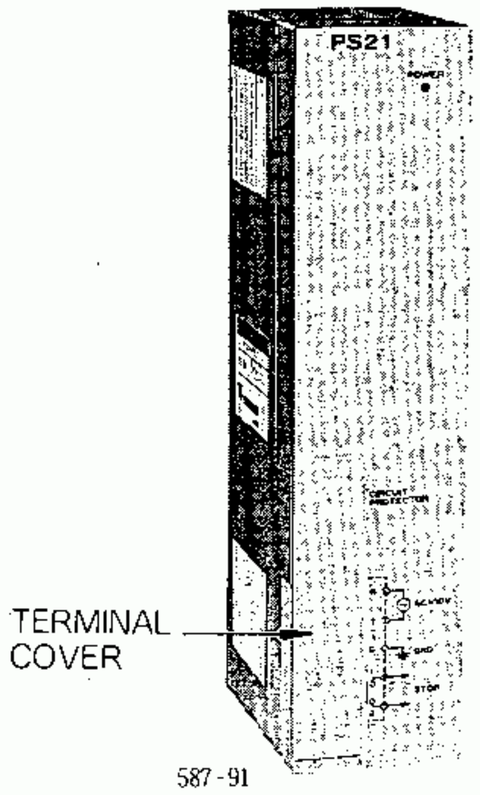


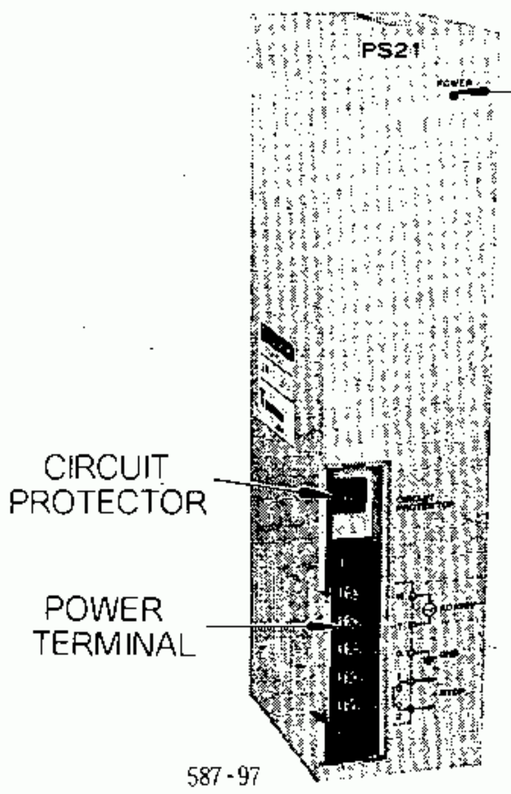
Fig. 3.3 Main Power Supply Module

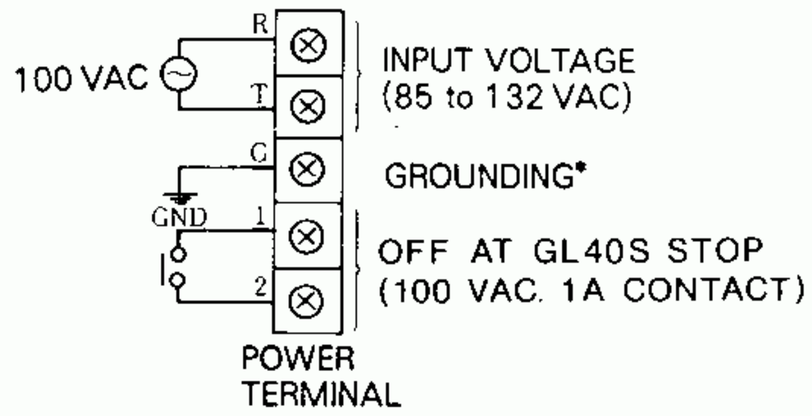
Table 3.5 Main Power Supply Module Specifications

Items	Specifications
Type	JRMSP-PS40
Function	DC power supply for a CPU module, function modules (COMM, etc.), and I/O (CH1 rack 1) modules
Input Voltage	Single-phase 85-132 VAC, 47.5-63 Hz, 100VA
Transient Input Voltage	0-154 VAC (10 ms)
Inrush Current	30 A (peak) or less
Leakage Current	1 mA or less
Fuse	Glass tube fuse (5 A)
Indicating Lamp	POWER: Lights when power supply is proper.
Monitoring Contact	STOP: ON at GL40S running, OFF at GL40S stop
Mounting Location	On mounting base MB40 (CPU base)
Dimensions in mm	60 (W) × 250 (H) × 94 (D)
Approx Weight	0.8kg

(2) Auxiliary Power Supply Module







*Grounding resistance:
100Ω or less

Fig. 3.4 Auxiliary Power Supply Module

Table 3.6 Auxiliary Power Supply Module Specifications

Items	Specifications
Type	JRMSP-PS21/PS22/PS22A
Function	DC power supply for I/O buffer module and I/O modules.
Input Voltage	Single-phase 85-132 VAC, 47-63 Hz, 70 VA (PS21), 100 VA (PS22/PS22A)
Transient Input Voltage	0-154 VAC (10 ms)
Inrush Current	30 A (peak) or less
Leakage Current	0.2 mA or less
Fuse	Circuit protector (3A) (Only for PS21)
Indicating Lamp	POWER: Lights when power supply is proper.
Monitoring Contact	STOP : ON at GL40S running, OFF at GL40S stop
Mounting Location	On mounting base MB22
Dimensions in mm	60 (W) × 250 (H) × 94 (D)
Approx Weight	0.7kg

3.2.4 Communication Module

(1) Expanding Communication Module (COMM)

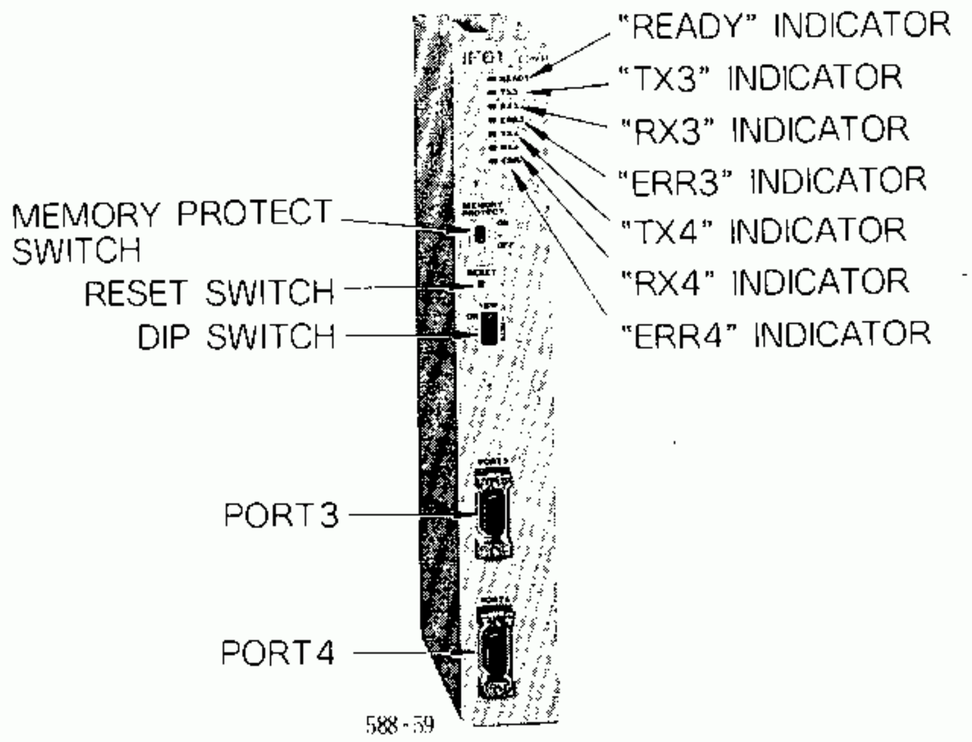


Fig. 3.5 COMM Module

Table 3.7 COMM Module Specifications

Items		Specifications
Type		JAMSC-IF61
Function		For more communication with P150 and a computer using 2 MEMOBUS ports (slaves).
Communication Port	No. of Ports	2 ports per module
	Communication Specification	EIA RS-232C
	Baud Rate	19200/9600/4800/2400/1200/600/300/150
	Data Bits	7 or 8 bits
	Parity	Even, odd or non
	Stop Bits	1 or 2 bits
	Protocol	MEMOBUS protocol
	Transmission Check	CRC-16 or LRC
	Connector	D-SUB 9 pin
Indicating Lamp (excluding RAP)		• READY: Lights when COMM module is proper. • TX3: Lights at port 3 transmitting. • RX3: Lights at port 3 receiving. • ERR3: Lights at port 3 communication error. • TX4: Lights at port 4 transmitting. • RX4: Lights at port 4 receiving. • ERR4: Lights at port 4 communication error.
Mounting Location		On mounting base MB40 (CPU base)
Dimensions in mm		37.5 (W) × 250 (H) × 94 (D)
Approx Weight		0.5 kg

Note
1. For dip switch setting, refer to Par.9.2.
2. JAMSC-IF61 type module can't be used with JAMSC-IF41A type module.

(2) Expanding Communication Module with RAP Port (COMM)

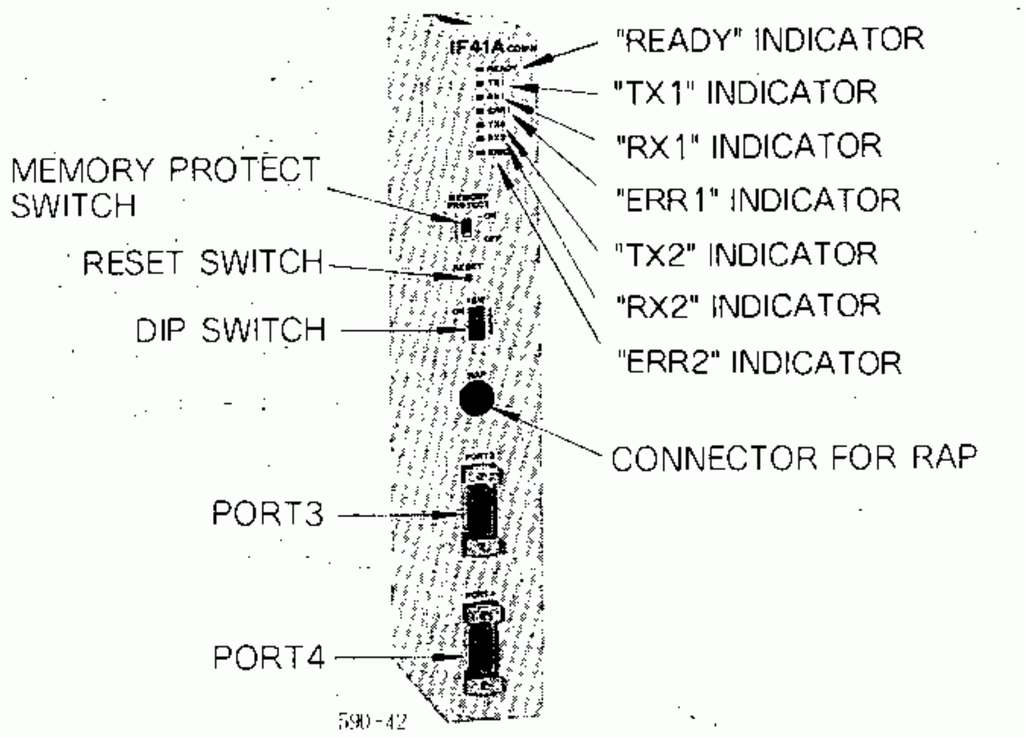


Fig. 3.6 COMM Module with RAP Port

Table 3.8 COMM Module (with RAP Port) Specifications

Items		Specifications
Type		JAMSC-IF41A
Function		For more communication with P150 and a computer using 2 MEMOBUS ports (slaves). RAP port can be mounted on the front.
Communication Port	No. of Ports	2 ports per module
	Communication Specification	EIA RS-232C
	Baud Rate	19200/9600/4800/2400/1200/600/300/150
	Data Bits	7 or 8 bits
	Parity	Even, odd or non
	Stop Bits	1 or 2 bits
	Protocol	MEMOBUS protocol
	Transmission Check	CRC-16 or LRC
	Connector	D-SUB 9 pin
Indicating Lamp (excluding RAP)		• READY: Lights when COMM module is proper. • TX1: Lights at port 3 transmitting. • RX1: Lights at port 3 receiving. • ERR1: Lights at port 3 communication error. • TX2: Lights at port 4 transmitting. • RX2: Lights at port 4 receiving. • ERR2: Lights at port 4 communication error.
Mounting Location		On mounting base MB40 (CPU base)
Dimensions in mm		37.5 (W) × 250 (H) × 94 (D)
Approx Weight		0.5 kg

Note 1. For dip switch setting, refer to Par. 9.2.
2. JAMSC-IF41A type module can't be used with JAMSC-IF61 type module.

3.2.5 PC Link Module

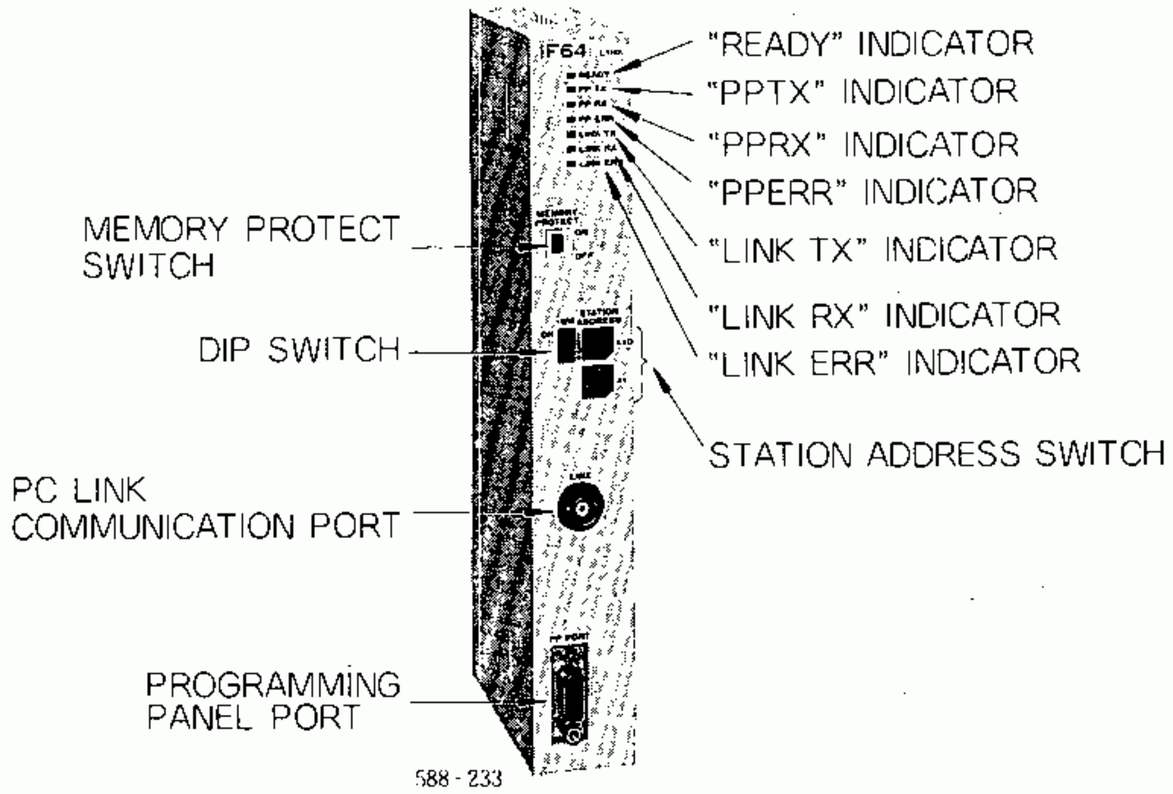


Fig. 3.7 PC Link Module

Table 3.9 PC Link Module Specifications

Items		Specifications
Type		JAMSC-IF64
Function		Performs mutual connections among PCs by the use of dedicated references.
PC Link	Topology (Net state)	Bus
	Transmission media	Coaxial cable
	Transmission method	Baseband (Manchester coding)
	Data baud rate	500 kbps/1 Mbps/2 Mbps/4 Mbps (Selected by SW_1)
	Max. cable length	1 km (but depending on transmission speed and used cable)
	Max. No. of station	32 stations
	Troubleshooting	• Failure station: Automatically disconnected from the line. • Recovered station: Automatically connected to the line.
Communication Port	No. of Ports	1
	Communication Specification	EIA RS-232C
	Baud Rate	9600 bauds
	Data Bits	8 bits
	Stop Bits	1 bit
	Parity	Even
	Protocol	MEMOBUS protocol
	Connector	D-SUB 9-pin
Indicating Lamp		READY: Lights when the module is proper. PP TX: Lights at PP port transmitting. PP RX: Lights at PP port receiving. PP ERR: Lights at PP port communication error. LINK TX: Lights at LINK port transmitting. LINK RX: Lights at LINK port receiving. LINK ERR: Lights at LINK port communication error.
Consumed Current		5 VDC 1.3 A
Mounting Location		On mounting base MB40 (CPU base)
Dimensions in mm		37.5 (W) × 250 (H) × 94 (D)
Approx Weight		0.6 kg

3.2.6 Servo Interface Module (IF66)

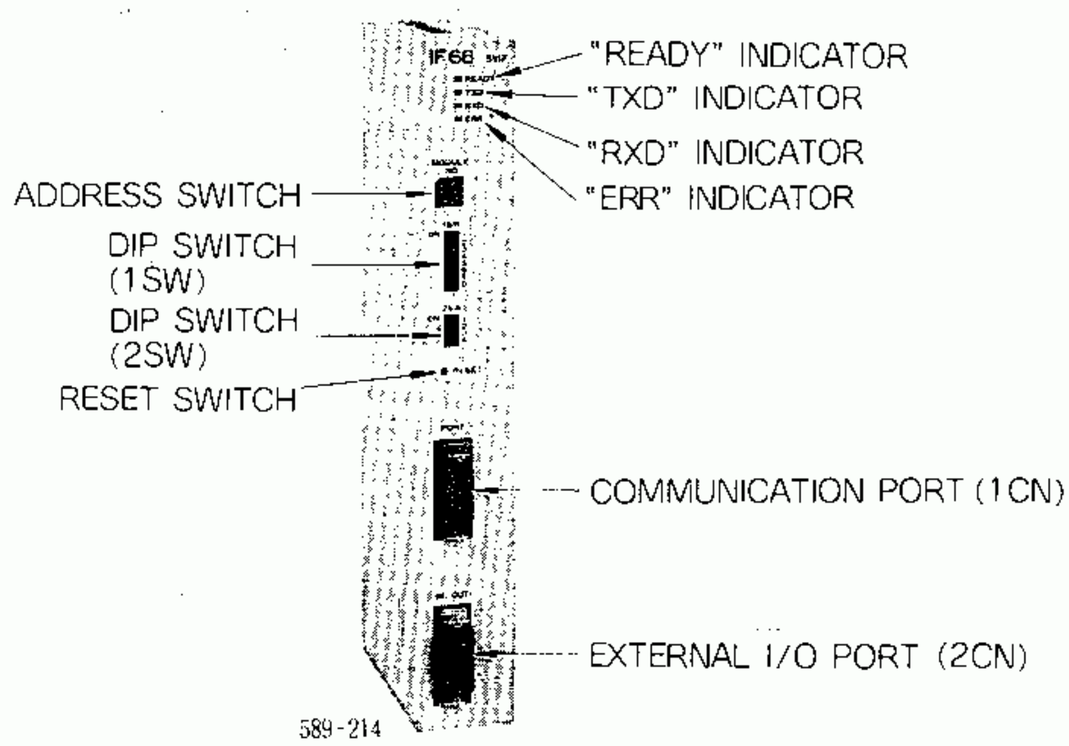


Fig. 3.8 · Servo Interface Module

Table 3.10 Servo Interface Modules Specifications

Items		Specifications
Type		JAMSC-IF66
Funcition		• Connected to Servopack CACR-HR □□ BA's, this is necessary for realizing motion control. • Interfaces the CPU module with Servopack. • One IF66 module is connectable to a max. of 8 CACR-HR □□ BA's.
Communication Port (1CN)	Communication Specification	EIA RS-422
	Communication System	Half-duplex asynchronous system
	Baud Rate	9600 bauds
	Bit Structure	JIS 7-bit system (Total 10 bits : Start 1 bit, data 7 bits, even parity 1 bit, stop 1 bit)
	Transmission Distance	Max. 30 m
	Error Check	Parity (Even), framing, BCC
External I/O (2CN)	Input	Servo alarm input (8 wires) Servo in operation input (8 wires)
	Output	IF66 normal output (1 wire)
Switch Setting	Address Switch	Designation of IF66 module address. 0 to 3 settable. Inhibition of overlapped use.
	Dip Switch (1 SW)	Setting of presence/absence of connected servo axis. [1SW-1: ON ··· Axis No. 1 servo present to 1SW-8: ON ··· Axis No. 8 servo present]
	Dip Switch (2 SW)	2SW-1: ON ··· External input signal selected. 2SW-2: ——— Reserved for the future. 2SW-3: ——— 2SW-4: ON ··· Test mode selected.
Indicating Lamp		READY: Lights when IF66 is normal TXD: Lights at data transmitting RXD: Lights at data receiving (Red LED lights at receiving error) ERR: Lights at ROM and RAM check error, WDT error, and system reset.
Mounting Location		On mounting base MB40
Dimensions in mm		37.5 (W) × 250 (H) × 94 (D)
Approx Weight		0.6 kg

3.2.7 I/O Buffer Module

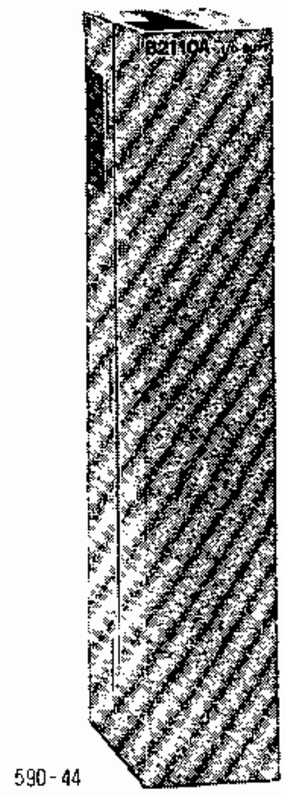


Fig. 3.9 I/O Buffer Module

Table 3.11 I/O Buffer Module Specifications

Items	Specifications
Type	JAMSC-B2110A
Function	• I/O bus buffer • For use of rack 2, 3 or 4
Connector	2 connectors for cables between racks (W20-1, -2)
Mounting Location	On mounting base MB22.
Dimensions in mm	46.3 (W) × 250 (H) × 94 (D)
Approx Weight	0.4kg

3.2.8 I/O Module

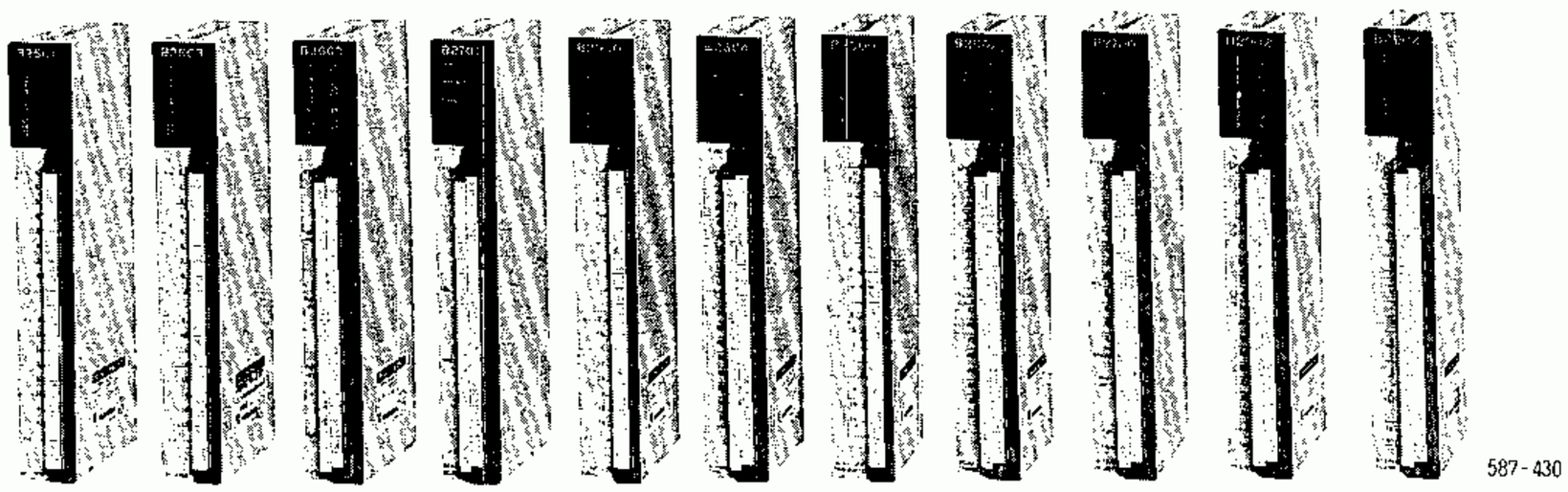


Fig. 3.10 I/O Modules

3.2.8 I/O Module (Cont'd)

Table 3.12 I/O Module Specifications

Modules		Items	Type JAMSC-	Voltage	Current	Input Impedance	External Power Supply	Maximum Response Time	No. of I/O's
Input	AC	100 V	B2501	100 V	10 mA	10 kΩ (at 50 Hz)	—	OFF → ON 15 ms ON → OFF 25 ms	16
		200 V	B2503	200 V	10 mA	20 kΩ (at 50 Hz)	—		16
		100 V	B2505	100 V	10 mA	10 kΩ (at 50 Hz)	—		32
		200 V	B2507	200 V	10 mA	20 kΩ (at 50 Hz)	—		32
	DC	12/24 V	B2601	12/24 VDC	5/10 mA	2.4 kΩ (at 50 Hz)	—	OFF → ON 5 ms	16
		48 V	B2611	48 VDC	9.4 mA	5 kΩ (at 50 Hz)	—		16
		12/24 V	B2603	12/24 VDC	5/10 mA	2.4 kΩ (at 50 Hz)	—	OFF → ON 5 ms ON → OFF 15 ms	32
		12/24 V	B2605	12/24 VDC	2.5/5 mA	4.7 kΩ	—		64
		5 V	B2607	5/12 VDC	4/11 mA	1.2 kΩ	—	0.5 ms	32
	Register		B2701	12/24 VDC	8 mA/24 VDC	2.4 kΩ	—	—	8
	Analog		B2703	0 to 10 V	—	—	—	—	8
			B2733	−10 to +10 V	—	—	—	—	8
			B2734	1 to 5 V	4 to 20 mA	—	—	—	8
Output	AC	100/200 V	B2500	100/200 VAC	1 A per output 3 A per 8 outputs	—	With CR, varistor. Fuse: 7.5 A per 8 outputs	OFF → ON 10 ms ON → OFF 15 ms	16
		100/200 V	B2504	100/200 VAC	0.3 A per output 1.2 A per 8 outputs	—	With CR. Fuse: 5 A per 16 outputs		32
	DC	12/24 V	B2600	12/24 VDC	2 A per output 5 A per 8 outputs	—	—	1 ms	16
		12/24 V	B2602	12/24 VDC	0.3 A per output 0.6 A per 4 outputs	—	—	1 ms	32
		12/24 V	B2604	12/24 VDC	0.1 A per output 0.4 A per 8 outputs	—	—	1 ms	64
		5 V	B2606	5/12 VDC	20 mA per output	—	—	1 ms	32
	Relay Contact		B2902	• 220 VAC, 1 A (Induction load, PF: 0.4) • 110 VAC, 1.2 A (Induction load, PF: 0.4) • 24 VDC, 1A (Induction load, TM†: 15 ms)		—	Relay coil voltage: 24 VDC ± 10% Min. operating voltage, current: 5 V, 10 mA	OFF → ON 10 ms ON → OFF 15 ms	32
			B2904	• 220 VAC, 0.5 A (Induction load, PF: 0.6) • 110 VDC, 0.3 A (Induction load, TM†: 40 ms)		—	Relay coil voltage: 24 VDC ± 5%. Min. operating voltage, current: 5 V, 1 mA	5 ms	16
			B2914	• 220 VAC, 0.5 A (Induction load, PF : 0.6) • 110 VDC, 0.3 A (Induction load, TM†: 40 ms)		—	Relay coil voltage : 24 VDC ± 5% Min. operating voltage, current : 24 VDC, 10 mA	5 ms	16
	Register		B2700	12/24 VDC	100 mA per output	—	—	—	8
	Analog		B2702	0 to 10 V	—	—	2 kΩ min. load	1 ms	2
			B2712	0 to 5 V	—	—			2
			B2722	−5 to + 5 V	—	—			2
			B2732	−10 to + 10 V	—	—			2
			B2742	4 to 20 mA	—	—	200 to 600 Ω load	5 ms	2
Motion Control	Reversible Counter		B2801	—	—	—	With coincidence output	—	2
	Preset Counter		B2802	—	—	—	—	—	1
	Positioning		B2803	—	—	—	• Analog output • For absolute	—	1
			B2813	—	—	—	Pulse output	—	1
			B2823	—	—	—	For pulse motor, servomotor	—	1
	PID Module		B2800	—	—	—	PID control	—	1
	Instrumentation Module		B2705	—	—	—	Instrumentation input	—	4
	MEMOLINK Master		B2804	—	—	—	Connecting to I/O MEMOLINK of 1000 series is possible	—	1
	MEMOLINK Slave		B2805	—	—	—		—	1

† : Time constant

3.2.9 Mounting Base

(1) MB40 Mounting Base

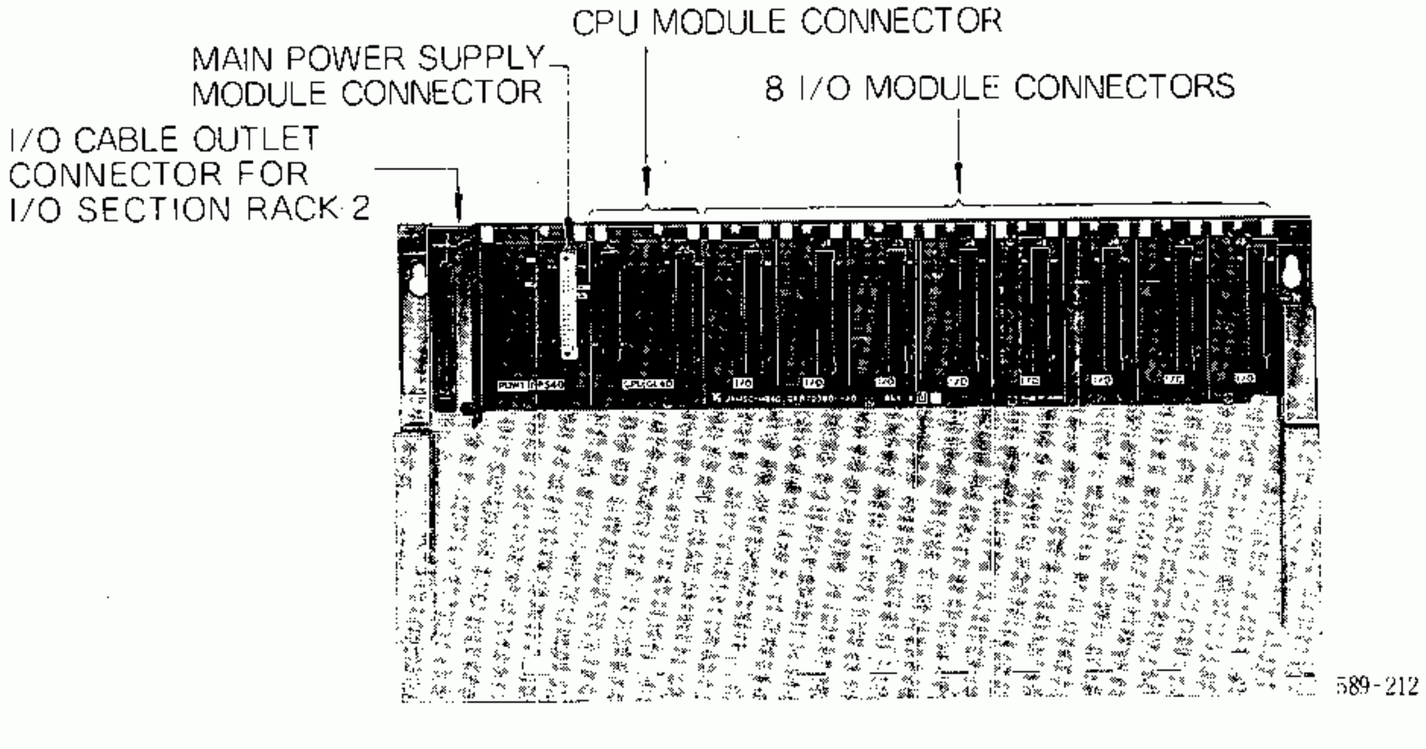


Fig. 3.11 MB40 Mounting Base

Table 3.13 Mounting Base MB40 Specifications

Items	Specifications
Type	JRMSI-MB40
Application	For mounting main power supply module, CPU module, COMM module, LINK module, SVIF module, and up to 7 I/O modules.
Dimensions in mm	480 (W) × 250 (H) × 21 (D)
Approx Weight	1.4 kg

(2) MB22 Mounting Base

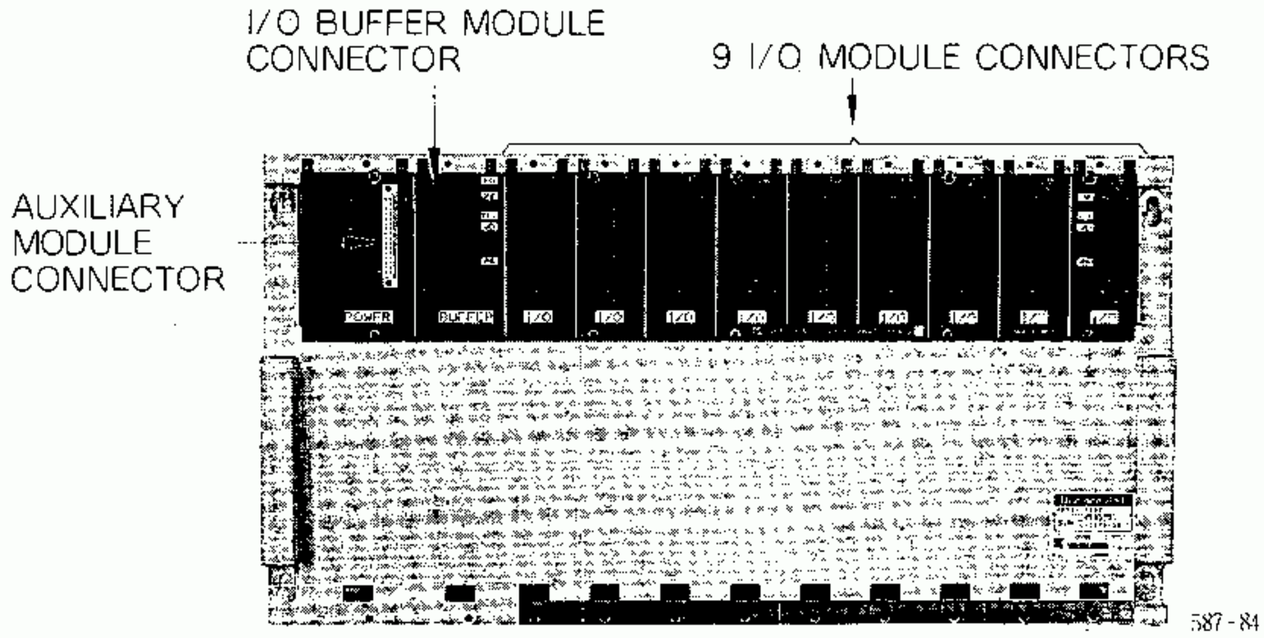


Fig. 3.12 MB22 Mounting Base

Table 3.14 Mounting Base MB22 Specifications

Items	Specifications
Type	JRMSI-MB22
Application	• For I/O expansion • For mounting auxiliary power supply module, I/O buffer module and up to 9 I/O modules.
Dimensions in mm	480 (W) × 250 (H) × 21 (D)
Approx Weight	1.3 kg

3.2.10 I/O Cable

The following types of cables are available for connection across each mounting base, respectively.

Table 3.15 I/O Cable Specifications

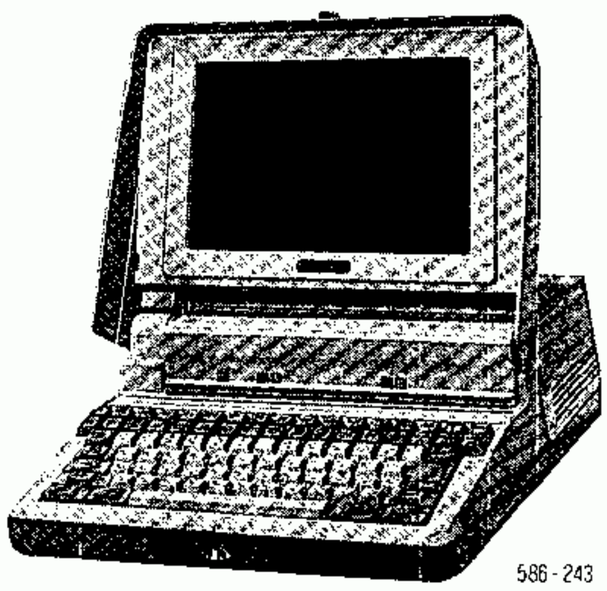
Items	Specifications	
Type	JZMSZ-W20-1	JZMSZ-W20-2
Length	0.5 m	1.5 m
Application	Used for connecting across each mounting base, respectively.	

3.3 PERIPHERAL MODULE

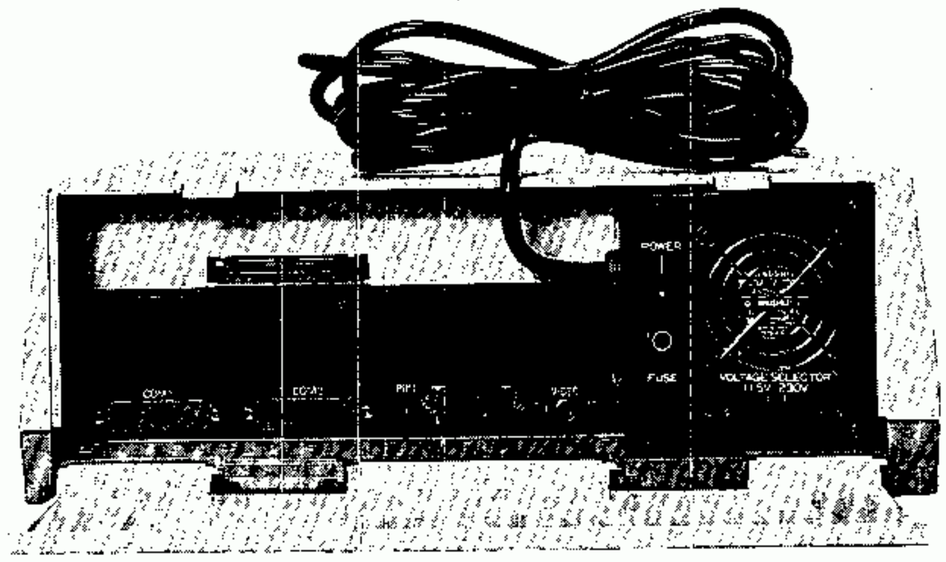
3.3.1 Programming Panel

(1) P150 Programming Panel

For details of the P150 handling, refer to DESCRIPTIVE INFORMATION "P150 Programming Panel User's Manual" :



(a) Front



(b) Rear

Fig. 3.13 P150 Programming Panel

Table 3.16 P150 Programming Panel Specifications

Items		Specifications
Type		DISCT-P150-10 (Standard keyboard – raised key) DISCT-P150-11 (Membrane keyboard – flush key)
Functions *		• Programming, adding, altering and deleting of logic and data • Logic entering, logic display • Load, dump and verify functions • Definition of system configuration • I/O allocation • Various monitorings, file control
Attachments	Graphic Display	Plasma display: 640 dot × 400 dot
	Keyboard	Label keys, function keyboard, numeric keyboard, ASCII keyboard, cursor control keys
	Floppy Disk Drive	Two 3.5-inch floppy disk drive
	Video	• Monitor type: Black and white, raster scan CRT (Type PC 8841 made by NEC Corp. is available.)
	Communication Port †	•One parallel port (made by Centronics Data Computer Corp.) • Two EIA RS-232C ports
Connection of P150 to Communication Module		Type W1015-T1 cable (length: 5 m) or W1015-T2 cable (length: 15 m)
Standard Specifications	Power	85 to 132 VAC/195 to 265 VAC, single phase, changeover at 50/60 Hz (47.5 to 63 Hz)
	Dissipated Power	120 VA
	Ambient Temperature	+5℃ to +40℃
	Storage Temperature	−20℃ to +60℃
	Humidity	20% to 80% relative (non-condensing)
	Atmosphere	Free from explosive, inflammable, corrosive gases, and dust.
	Grounding	Chassis grounding line is connected to mainframe grounding line via connecting cable of CPU module.
	Approx Weight	9 kg

* A control program in a 3.5-inch floppy disk must be loaded in the P150.
† A ladder diagram can be printed out through a printer connected to the P150.

<3.5-inch Floppy Disk>

Before using the P150 programming panel, a control program in a 3.5-inch floppy disk must be loaded in the P150. Select one of the following 3.5-inch floppy disks suitable for the intended use.

① GL40S Programmer FD (F40S-E001)

This is used to make a program and control load, dump, verify and file functions with P150, and also used to perform I/O allocation.

② GL40S Ladder Lister FD (F40S-E002)

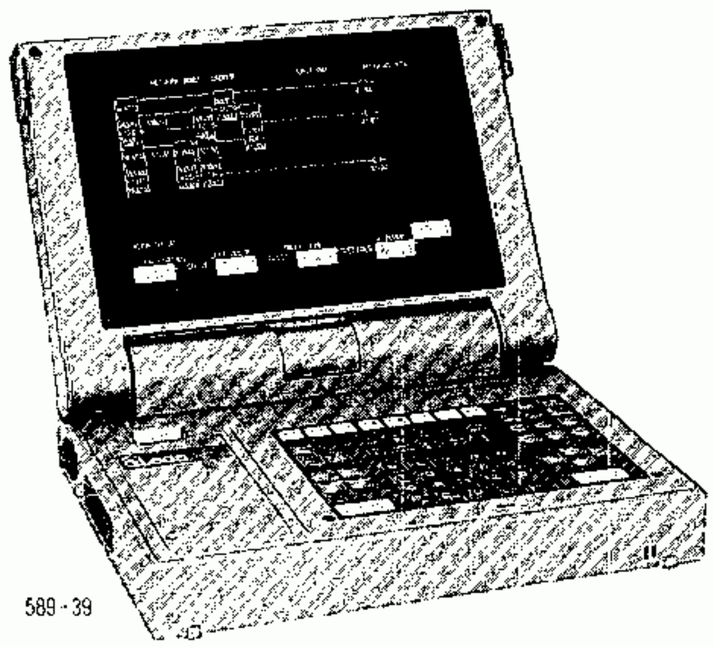
This is used to print out a ladder diagram through a printer connected to the P150.

③ Blank FD (F150-000)

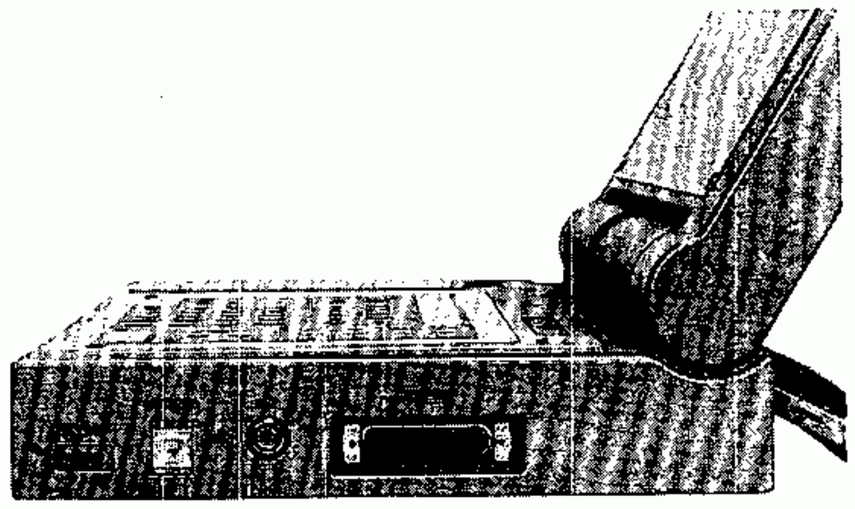
A blank floppy disk is used to record the ladder circuits stored in the CPU module memory by using a programmer floppy disk. The contents of the registers and I/O allocation are recorded together.

(2) P140 Programming Panel

For details of the P140 handling, refer to DESCRIPTIVE INFORMATION "P140 Programming Panel User's Manual":



(a) Front



(b) Rear

Fig. 3.14 P140 Programming Panel

Table 3.17 P140 Programming Panel Specifications

Items		Specifications
Type		DISCT-P140-20
Functions*		• Programming, adding, altering and deleting of logic and data • Logic entering, logic display • Load, dump and verify functions • Definition of system configuration • I/O allocation • Various monitorings, file control
Attachments	Graphic Display	Liquid crystal display: 640 dot × 400 dot
	ROM Pack	Built-in system program, for each type.
	Flat Key	61 keys
	Communication Port	EIA RS-232C port × 1
	FDD Interface	Connects to peripheral FDD unit (FD400)
	JIS Keyboard Interface	Connects to JIS Keyboard (KB400)
Connection of P140 to Communication Module		Type W1015-T1 cable (length: 5 m) or W1015-T2 cable (length: 15 m)
Standard Specifications	Power	85 to 132 VAC, single phase, changeover at 50/60 Hz (47.5 to 63 Hz)
	Dissipated Power	50 VA
	Ambient Temperature	+5℃ to +40℃
	Storage Temperature	−20℃ to +60℃
	Humidity	20% to 80% relative (non-condensing)
	Atmosphere	Free from explosive, inflammable, corrosive gases, and dust.
	Grounding	Chassis grounding line is connected to mainframe grounding line via connecting cable to CPU module.

* Peripheral FDD unit (FD400) and standard keyboard-raised key are needed.

<Specifications of ROM Pack>

The ROM pack is a memory module containing programming software for the target PC. Select a ROM pack depending on the target model.

(1) Specifications

Table 3.18 Specifications

Items	Specifications
Type No.	P40S-E001 Memocon-SC GL40S
Memory	ROM 640 kB, EEROM 2 kB

(2) Handling ROM Pack

① Do not touch the connector pins on the back of the ROM pack.
② Turn off the power before replacing the ROM pack.
③ Be sure the connector is fully inserted.

<Specifications of Programming Keyboard>

The programming keyboard is used for entering comments and file names, and for loading or saving a user program to P140 using and FDD unit. This keyboard is also used for the ACGC 400 series.

Table 3.19 Specifications

Items	Specifications
Type No.	DISCT-KB400
Interface	Start-stop synchronization (9600 baud/sec, 8-bit code with no parity bit)
Dimensions in mm	460 (W) × 36 (H) × 167 (D)

<Specifications of FD400>

The FD400 is the external memory unit for the P140 and is used for loading and saving user programs.

Table 3.20 Specifications

Items	Specifications
Type No.	DISCT-FD400
Drive	YD-686C
Medium	3.5-inch floppy disk (2DD/2HD)
Capacity	1.6 or 1.0 M byte (not formatted)
Format	MS-DOS
Dimensions in mm	150 (W) × 100 (H) × 220 (D)
Approx Weight	2 kg

3.3.2 Register Access Panel (RAP)

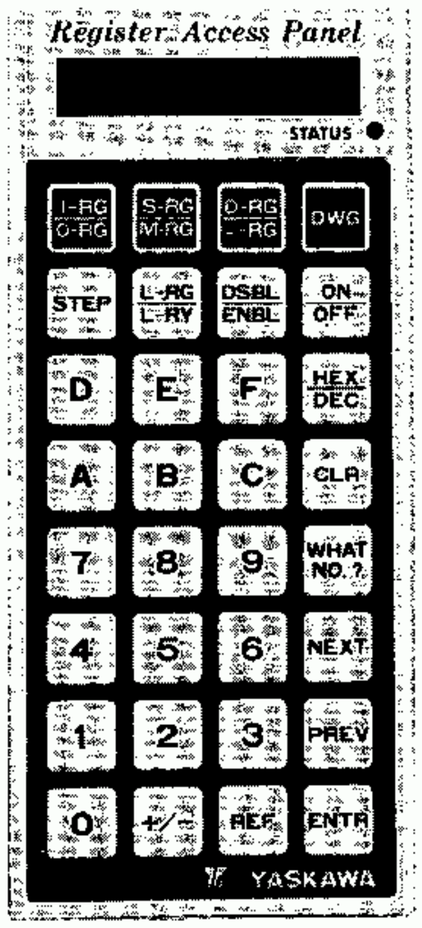


Fig. 3.15 Register Access Panel

Table 3.21 RAP Specifications

Items	Specifications
Type	DISCT-IF69
Function	• For coil or input relay: monitor ON/OFF status, disabling (forced ON/OFF). • Displaying or altering register contents. • Setting or displaying communication parameters. • Monitoring ON/OFF Status of coil or input relay by status indication LED. • 1-scan pulse monitoring available.
Indication	• 16-segment, 8-digit alphanumeric LED. • Status indication LED
Mounting Location	On expanding communication module with RAP port.
Dimensions in mm	70 (W) × 155 (H) × 19.5 (D)
Approx Weight	0.3 kg

SECTION 4
IMPORTANT MACHINE CONCEPTS

4.1 USER PROGRAM CONFIGURATION

User programs include three sizes according to the type of the arithmetic unit.

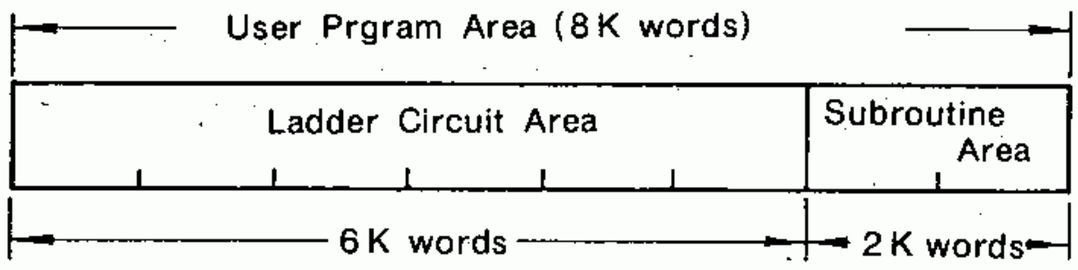
GL40S1 ---2k words (24 bits/word)
GL40S2 ---4k words (24 bits/word)
GL40S3 ---8k words (24 bits/word)

These numbers show the area that stores sequence programs consisting mainly of relays, timers, counters, and stepping switches and data processing programs such as arithmetic operation, data transmission, and matrices. The area that stores subroutine circuits must be separated from the area that stores normal ladder circuits. These areas are allocated beforehand with the programming panel (P150 and P140).
(For the operating method, refer to Programming Panel Manual SIE-815 -15.2, 15.3.)

The allocation of the subroutine circuit area is made for every 0.25k words. The size of ladder circuits is reduced automatically by the subroutine circuit area.

In the case of GL40S3, for example, if subroutine circuit allocation is set at 2k words, the ladder circuit area becomes 8k words - 2k words = 6k words.

If subroutine circuits are not necessary and they are not used, setting subroutine circuit allocation at 0k words will make the ladder circuit area 8k words.



If several parts excute the same operation in a ladder circuit, this part is made a subroutine circuit, thus saving memory.

4.2 NETWORKS

The GL40S program is composed in units of network (Only one network for transition condition circuit). The multi-node format allows for up to ten elements of the program in each horizontal rung of the ladder diagram. Up to seven of these rungs can be combined into a network of relay contacts and other programming elements (timers, counters, etc.); each network can have up to seven coils placed at the extreme right of the network.

The networks are stored in the memory of the CPU module in the order of the network numbers.

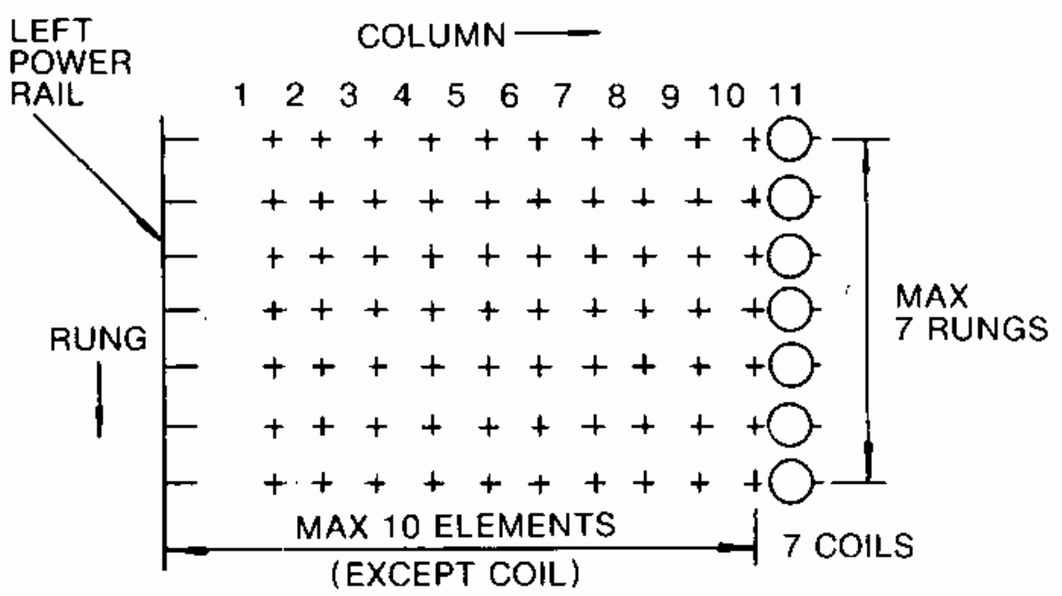


Fig. 4.1 Networks

4.3 CONTROLLER REFERENCE NUMBERS

Throughout the programming of the GL40S Controller, five-digit reference numbers are utilized to build the user's logic. These references are divided into two broad categories: discretes and registers. Discrete references are used for individual items that can be either ON or OFF, such as limit switches, pushbuttons, relay contacts, motor starters, relay coils, solenoid valves, etc. Register references are used to store numerical values such as counts, times, analog values, etc.; all register references are three BCD digits long (maximum value 9999). Since there are four bits per BCD digit, registers can also be 16 bits of data.

Only nine types of references are required to program the GL40S Controller. Any specific reference can be used as many times as required by the particular application; there are no limitations on the number of times a reference is used.

Table 4.1 shows classification of reference numbers.

4.3 CONTROLLER REFERENCE NUMBERS (Cont'd)

Table 4.1 Range of Reference Numbers

Reference Number	Elements	Remarks
00001-00512	Output coils and their contacts	Available as internal coils.
00513-02047	Internal coils and their contacts	
02048	Battery monitoring contact	ON when battery voltage is proper.
D0001-D1024	Link coils and their contacts	
10001-10512	Input relays	
30001-30128	Input registers	
40001-40128	Output registers	Available as holding registers
40129-42048	Holding registers	42001-42032: for stepping control
R0001-R1024	Link registers	
42047	Constant sweep set value register	Available as holding registers when not used for constant sweep.
42048	Actual scan time register	
2YYXX	Stepping relay (YY: 01-32, XX: 01-99)	
MXXYZ	M code relay (XX: 50 or 51, Y: IF66 No. 1 to 4, Z: Axis No. 1 to 8)	
N00YZ	N code relay (Y: IF66 No., Z: Axis No. 1 to 8)	

(1) Coils (0XXXX)

Other than the coils used in a relay circuit, coils can be used for any results of processing such as timer, counter, arithmetic operations, data move, data convert, matrix, etc.

Coils are divided into two types: normal coil (not retentived when power is OFF) and latch coil (retentived even after power is OFF). They are divided into two groups from another standpoint. One is the internal coil (auxiliary relay) which is used only in a ladder circuit but not as an external output. The other is the output coil which drives an external device via an output module.

The coils are identified by specific coil numbers. The coil contacts (having the same reference numbers as the coils) may be used repeatedly in any network.

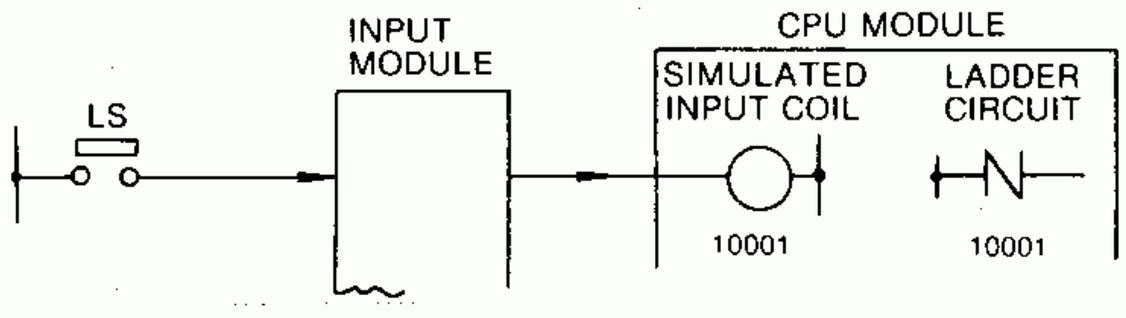
Note Coil 02048 cannot be used in a program since it is used as a battery monitor coil. However, its contact may be used repeatedly in a program.

(2) Link Coils (DXXXX)

The GL40S can link PCs (up to 32 PCs) by its PC link system to perform high speed data link. The ON/OFF state of the coil output by the link coil which is allocated to the local system by a GL40S can be referenced by another GL40S at a remote station through a link relay.

(3) Input Relays (1XXXX)

An input relay is an ON/OFF signal (discrete input signal) entered through an input module. It may be considered as the contacts of a simulated input coil stored in the CPU module. The contacts of an input coil (input relay) may be used repeatedly in any network. An input relay may be referenced repeatedly but cannot be altered.

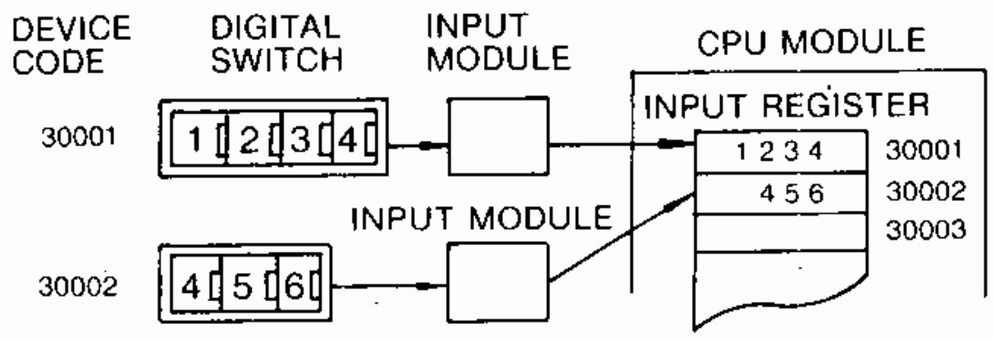


(4) Stepping Relays (2YYXX)

A stepping relay is an contact of stepping switch relay (ON/OFF signal) and there are 32 contacts. Simulated coil becomes ON/OFF according to the register (42001 to 42032).

(5) Input Registers (30XXX)

An input register is memory (16 bits) for temporarily storing a numeric signal sent from an external device such as a digital switch, card reader, A/D converter, or computer. Each input register is identified by a reference number beginning with 30XXX. The reference number may be considered as a device code given to the associated external device.



An input register stores 16-bit numeric data. Binary data input from an external device can be used as is, but BCD data must be converted to binary data by the operation fucntion BIN. If BCD data are used for an internal operation, the result will be incorrect.

Fig. 4.2 shows connection for numeric data input from the digital switch.

Note The contents of any input register is binary in the memory. It is possible to refer to the contents of any input register but impossible to alter them in the ladder circuit.

(6) Holding Registers (4XXXX)

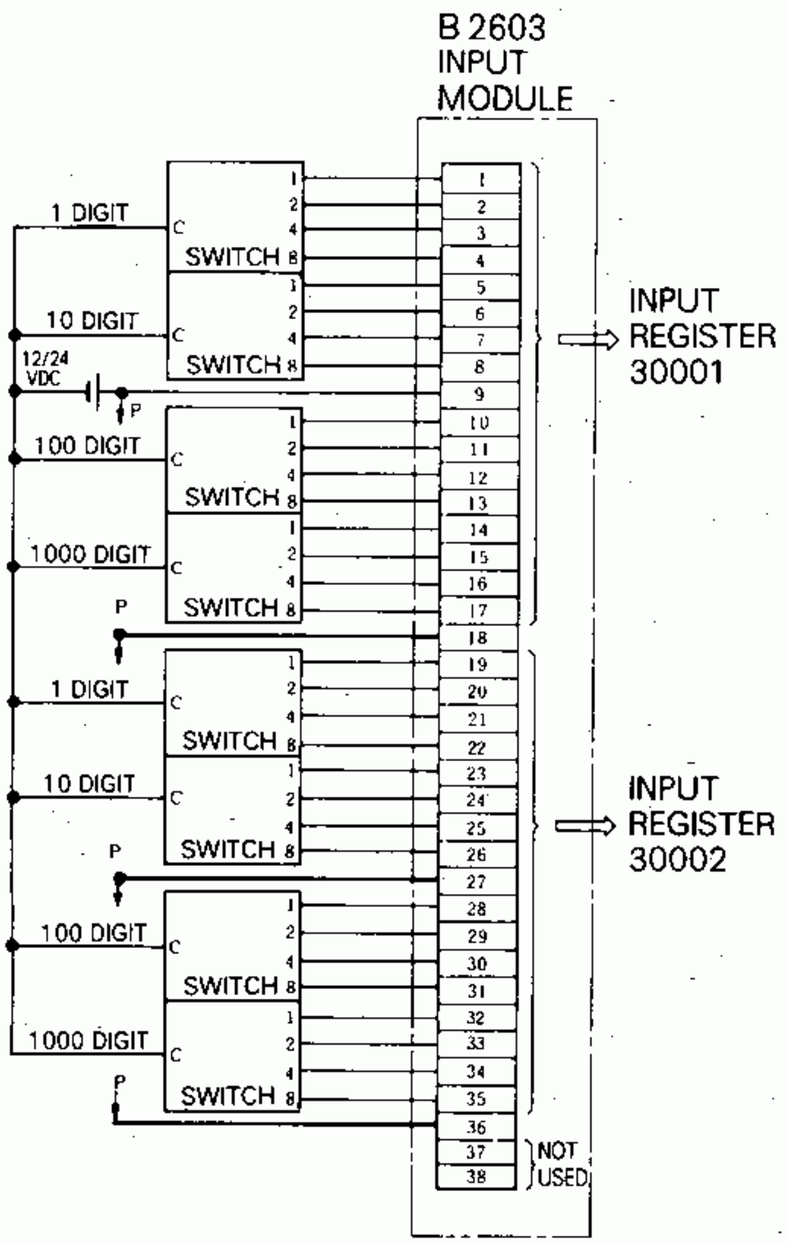
The holding registers (of 16 bits each) hold the preset values and current values of the timer and counter, constants and results of arithmetic operations, transferred data, converted data, matrix data, or other constants needed for processing. Each holding register is identified by a reference number beginning with 4XXXX.

A holding register holds numeric data of 16 bits in binary form. The contents of the holding registers can be referenced or altered by the ladder program, and can be held at power failure.

The contents of the holding register can be output to an external device via an output module if necessary. At this time, the holding register may be called an output register. If the external device to which data are to be output handles BCD data, the output data must be converted from binary to BCD.

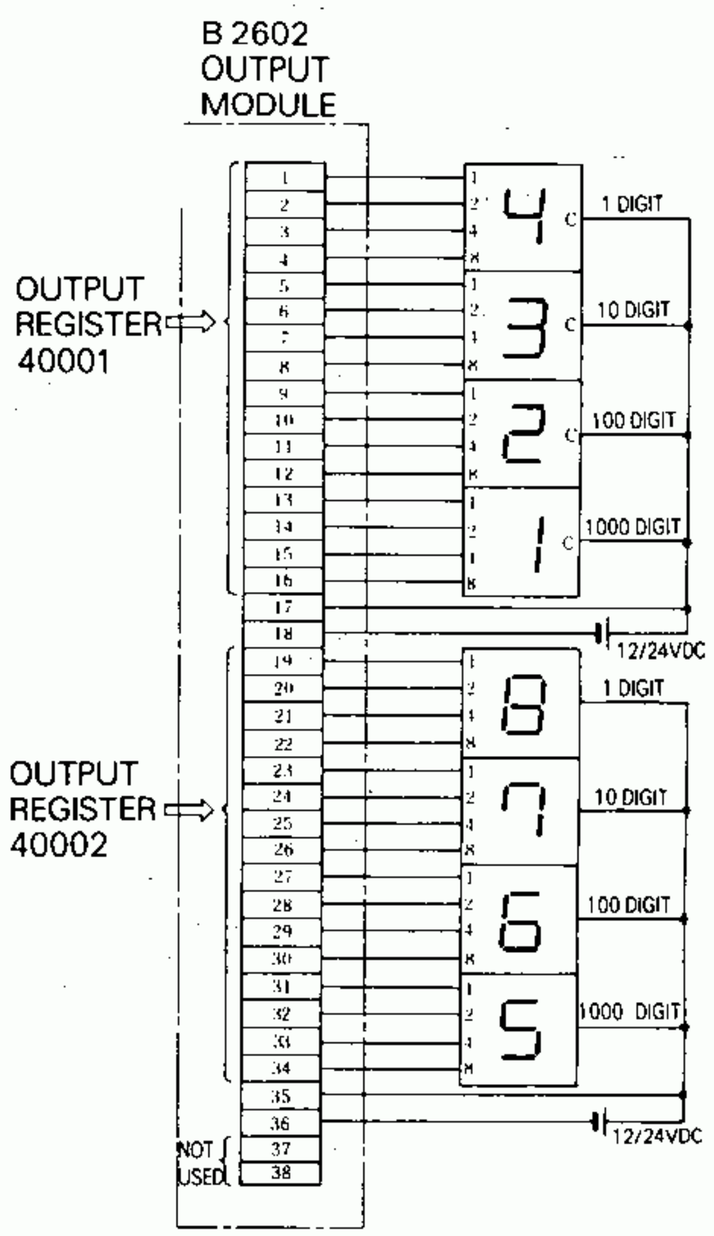
Fig. 4.3 shows connection for numeric data output to a BCD digital display device.

Note The contents of all the holding registers are binary.



Note To use for an operation, the data in the input register must be converted to binary.

Fig. 4.2 Sample Connection for Numeric Data Input



Note
1. Data must be converted to BCD before output.
2. The digital display device is assumed to be of 12/24VDC interface, negative logic, and BCD.

Fig. 4.3 Sample Connection for Numeric Data Output

(7) Link Registers (RXXXX)

The link registers, as well as the link coils, are used when a PC link system is built. Data stored in the link registers which are allocated at the local station by a GL40S can be referenced by another GL40S at a remote station. That is, the link registers allocated at the local station can be used as "write-enabled registers." and those allocated at a remote station can be used as "read register."

4.4 NUMERAL DATA NOTATION

Numeral data notation includes 4 types.

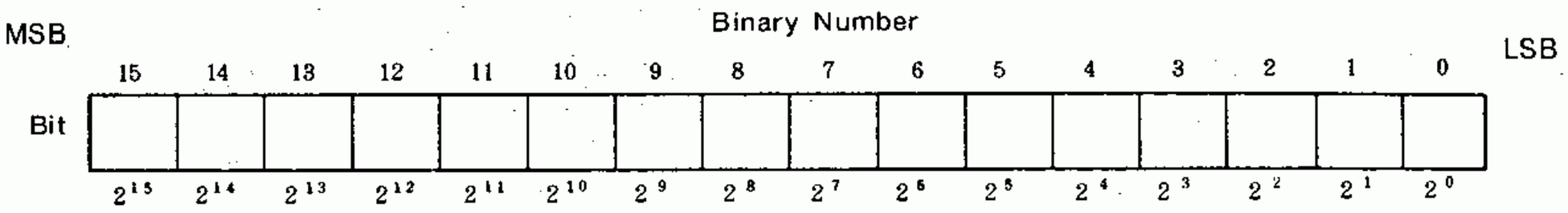
1. Decimal
2. Binary
3. BCD (Binary Coded Decimal)
4. Hexadecimal

4.4.1 Positive Integer Notation

Holding registers in GL40S use binary numbers represented by "1" and "0". Such numbers, however, are represented as decimal numbers on the programming panel. In the binary system, the first digit represents 2^0(=1), the second digit 2^1(=2), the third digit 2^2(=4) --- , thus the nth digit represents $2^{(n-1)}$.

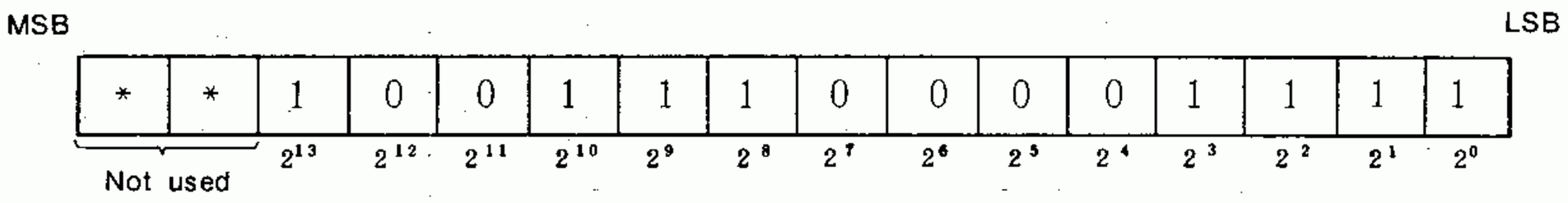
For example, since a decimal number 1111 is expanded as $2^{10} + 2^6 + 2^4 + 2^2 + 2^1 + 2^0$, it is represented by an 11-digit binary number 1 0 0 0 1 0 1 0 1 1 1 .

A register is composed of 16 bits.



The right end 2^0, the lowest position bit, is called LSB (Least Significant Bit). The left end 2^{15}, the highest position bit, is called MSB (Most Significant Bit).

Numeral data handled by GL40S are 4-digit decimal numbers (0 to 9999) for a register. Therefore, the 15th bit (2^{14}) and the 16th (2^{15}) are not used in the case of numeral data. The maximum number 9999 of 4-digit decimal numbers is represented by the following state of bits.



Numeral data use of 5-digit to 8-digit decimal numbers are stored by the use of 2 registers.

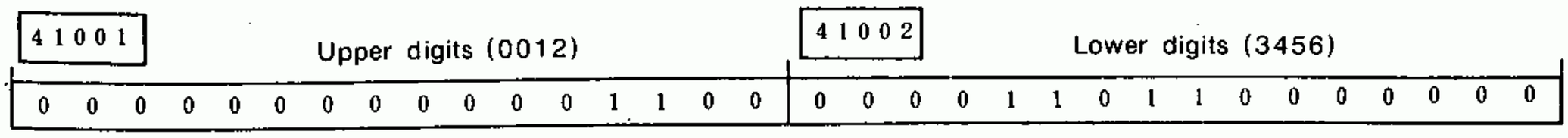
For example, a 6-digit number 123,456 is divided into 0012 and 3456, before being stored in 2 registers.

Holding register numeral

41001	41002
0 0 1 2	3 4 5 6

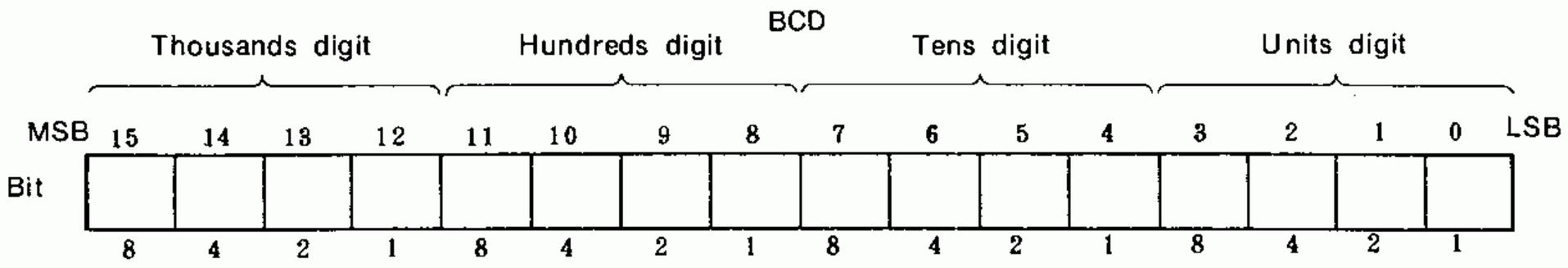
Upper 4 digits Lower 4 digits

This is represented by the following state of bits.

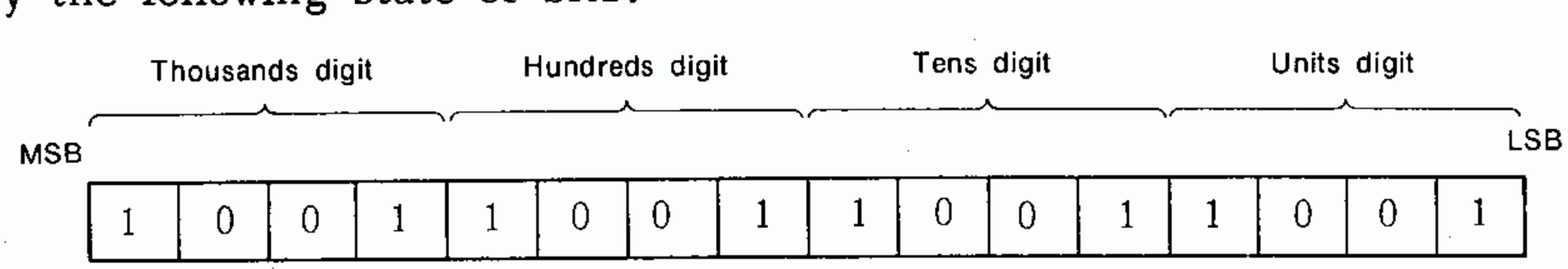


In the BCD system, on the contrary, 1-digit decimal numbers are made with 4-digit binary numbers to be combined to make 4-digit decimal numbers. Some digital switches and numeral indicators have the BCD code, and use BCD data as I/O signals. In this case, BCD inputs must be converted into binary numbers and binary outputs must be converted into BCD. In GL40S, BIN performs BCD→BIN conversion and BCD performs BIN→BCD conversion as one of its arithmetic functions.

The figure below shows the BCD representation.



The maximum number 9999 of 4-digit BCD numbers is represented by the following state of bits.



Next, 1-digit hexadecimal numbers are represented as 0 to 9, A, B, C, D, E, and F by the use of 4-digit binary numbers.

DECIMAL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HEXADECIMAL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
BINARY	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

4.4.1 Positive Integer Notation (Cont'd)

The maximum number FFFF of 4-digit hexadecimal numbers is represented by the following state of bits.

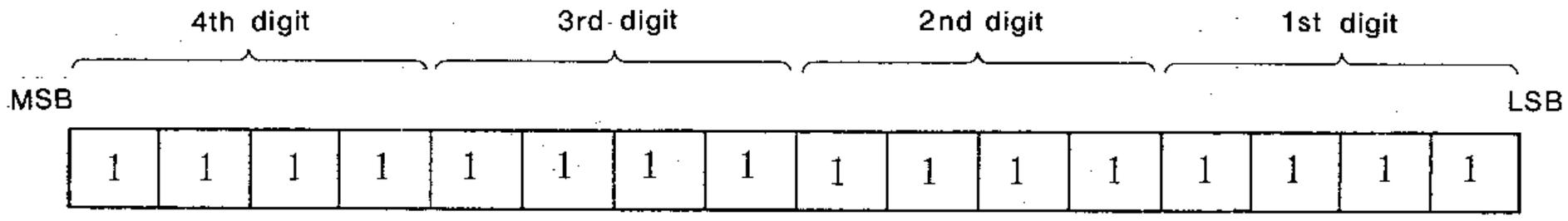


Table 4.2 Example of Decimal Numbers Represented in Binary, BCD, and Hexadecimal Systems

DECIMAL	BINARY	BCD	HEXADECIMAL
5	0000 0000 0000 0101	0000 0000 0000 0101	0005
12	0000 0000 0000 1100	0000 0000 0001 0010	000C
100	0000 0000 0110 0100	0000 0001 0000 0000	0064
2000	0000 0111 1101 0000	0010 0000 0000 0000	07D0

4.4.2 Sign Notation

GL40S performs negative numeral operations with the signed arithmetic function. To represent positive (+) or negative (-), the sign bit is provided. The sign bit "0" is understood to be positive (+) and the bit "1" to be negative (-). The sign bit is fixed at the MSB (Most Significant Bit) in 16 bits.

• In GL40S, the internal process of negative numbers is as follows:

① A negative decimal number entered from P150 programming panel is converted into a binary number, first regarding it as a positive number.

② 1 is set to the Most Significant Bit (MSB) of a converted binary number.

③ The numeral is stored in a designated reference number.

Example 1: When -100 is entered.

100 is converted into a binary number, 0000 0000 0110 0100

↓

1 is set to MSB, 1000 0000 0110 0100

↓

stored in a designated register.

Example 2: When -10,005,000 is stored in continuous registers.

40001	−1000
40002	5000

The lower part of the data is entered without a negative sign (-). The result of operations is processed in the same way. The lower part of the data is stored as a positive number.

Note

1. It is unacceptable to mix signed and unsigned arithmetic operations in an operation system. Different data structures might disrupt correct operations.
2. When any processing such as data conversion is performed to operands or numbers to be operated, 1 might stand at the MSB to make them negative numbers. Care should be taken to avoid this.

4.5 GL40S INTERNAL PROCESS

Fig. 4.4 shows the GL40S internal processing flow chart.

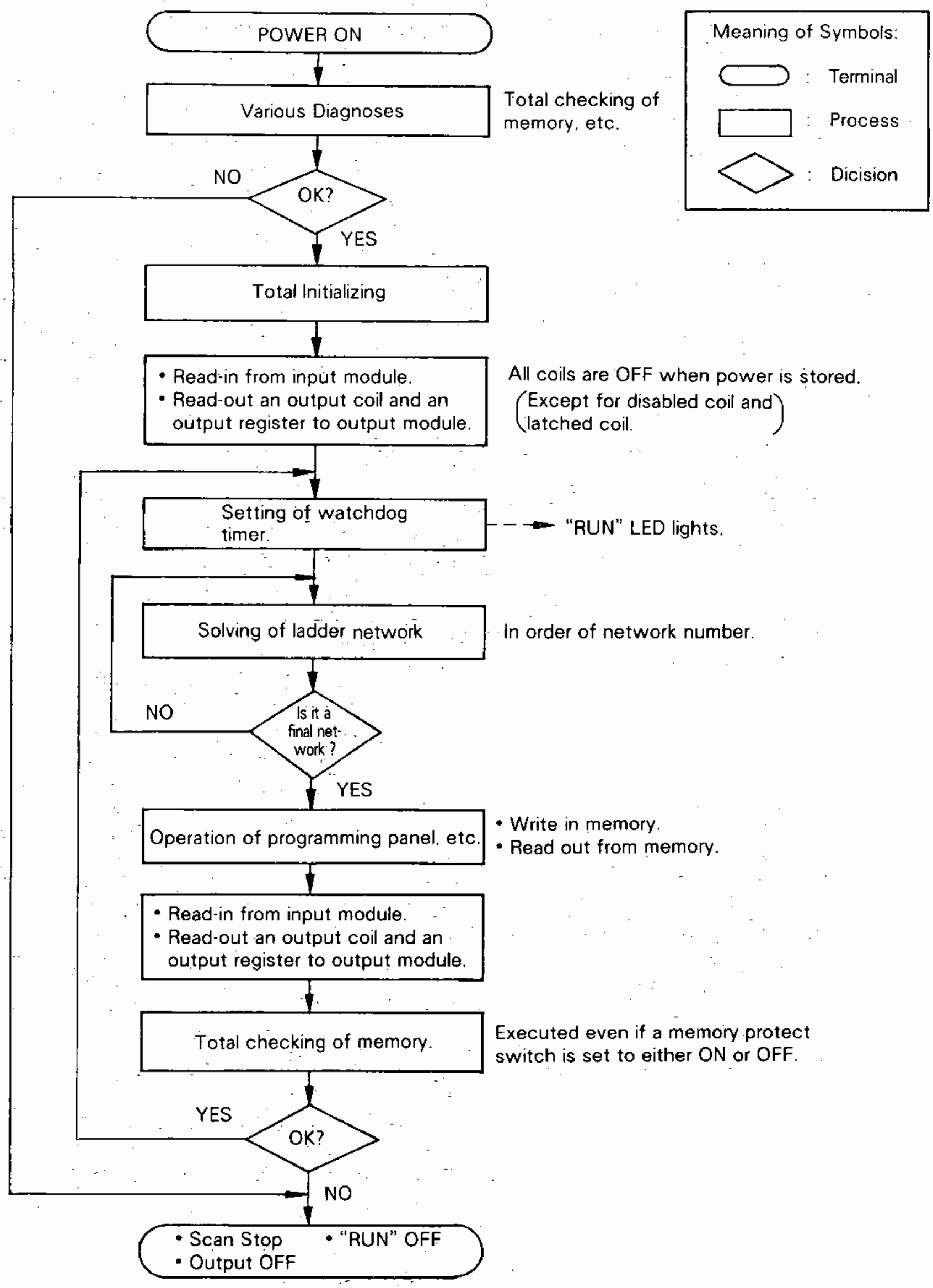


Fig. 4.4 GL40S Internal Processing Flow Chart

(1) Power-up Sequence

When power is turned on, the system checks the contents of the memory and, if normal, initializes all input relays and coils, etc.

Table 4.3 shows the initialized statuses of input relays and coils.

Table 4.3 Initialized Statuses of Input Relays and Coils

Elements	Initialized Status
Coil, Link Coils*	OFF except for latched coils and disabled coils
Input Relays Input Registers, Link Registers†, Link Coils†	Latest status except for disabling input relays
Latched Coils, Disabling Inputs, Disabling Coils	Status held immediately before power failure
Holding Registers, Link Registers*	Status held immediately before power failure

* Local station
† Remote station

The power-up sequence takes approximately 5 seconds.

(2) Scan Cycle

Upon completion of a power-up sequence, solving the ladder circuit is performed in order of network numbers beginning at network 1. The processing related to the programming panel and I/O processing are completed after the last network is solved, then the GL40S returns to network 1, and repeats the same procedure.

This is called a scan cycle operation, and the time required for one cycle is called scan time.

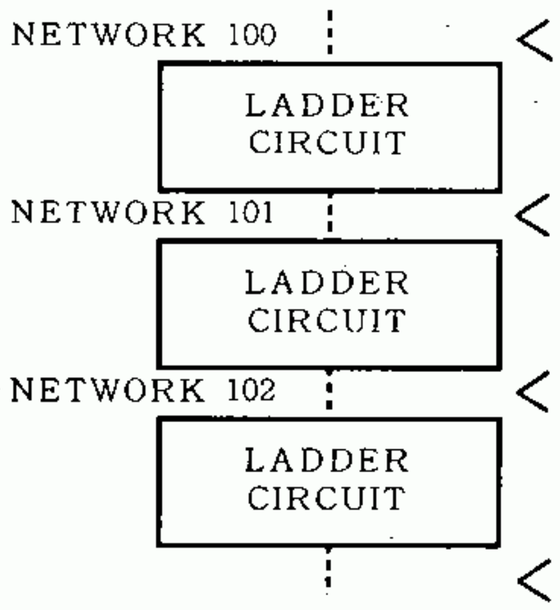
(3) Watchdog Timer

A watchdog timer is set at the beginning of every scanning cycle. The timer remains ON for approximately 500 ms after it has been set. If the timer has not been set within a certain time interval, the system stops scanning, determining that there is something wrong with the scanning. At this time, the RUN LED on the CPU module comes OFF and all outputs become OFF. This is one of the self-diagnostic functions.

(4) Network Number

The ladder program is stored in the program memory in units of the network. The number of elements in a network must be in a range of 7 lines × 10 columns + 1 column (coil). The designer should designate the boundary of the network. The networks are numbered serially (1, 2, 3, and so on). As the designer designates the boundaries of the networks, the system assigns network numbers to them automatically. There are no limitations adding or deleting elements in a network, adding or deleting networks, or inserting a new network between adjacent networks. When some networks are added or deleted, the system controls the network numbers automatically.

Note Network numbers may increase until the program memory becomes full.



Note The designer designates the boundaries of each network (shown with <). It is actually performed by pressing the START NEXT key on the programming panel.

Fig. 4.5 Networks and Their Delimitations

(5) Checksum

A serious error might occur if some contents of the memory, such as a user program or major system constants have been changed during operation and the changes cannot be detected. To prevent such problems, the system is provided with a self-diagnostic function called "Checksum." The total sum of the contents of the memory which should not be changed has been set. In every scanning cycle, the sum is calculated and compared with the previously setsum. If a discrepancy occurs, the GL40S stops scanning as with any error. Simultaneously, all outputs connected to the output module are turned off. Register and analog outputs hold value before stopping the scan.

Total check covers the following.

- **Ladder circuits**
- **I/O allocation table**

Note Total check is performed regardless of the memory protect switch position. When in OFF position, some ladder circuits may be altered through the programming panel and it will not be detected as an error by total check. Rather the total check sum is renewed and comparison will be made with reference to this new value.

4.6 SCANNING

4.6.1 Solving a Ladder Circuit

Each network is examined (solved) in order of column one to column ten and then to the coils. Within each column, the logic is solved from the top rung to the bottom rung of that column. The new results from each network (either coil status or register content) are immediately available for use by the next network or column. The scan is done by network number not by output coil number. Fig. 4.6 shows the sequence the GL40S solves networks. The scanning technique is essential to the operation of the GL40S controller and should be understood before proceeding. Table 4.4 shows the status of each element during scanning.

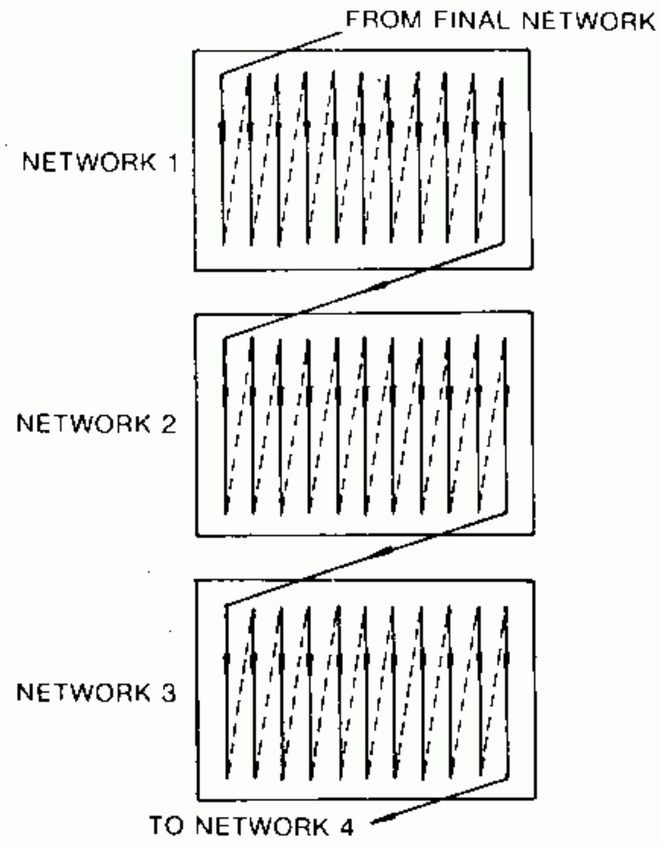


Fig. 4.6 Sequence of Solving Networks

Table 4.4 Status of Each Element During Scanning

Element	Status
Input Relays (Except for Disabling Operation Inputs), Input Registers	Latest status after power-up sequence. Then input stauts, updated just when inputs are read in, remain unchanged until the next scanning.
Coils (Except for Latch Coils and Disabling Operation Coils)	OFF after power-up sequence. A coil is turned on or off according to the result solved in the first scanning status remains until the next scanning.
Latched Coils, Step Relays	After power-up sequence, status before power failure are restored. During scanning, the same as with coils.
Disabling Operation Inputs, Disabling Operation Coils,	After power-up sequence, status before power failure are restored. Statuses are updated only when changed through the programmig panel at the end of scanning.
Holding Registers, Link Registers	After power-up sequence, status before power failure are restored. During scanning, the contents vary according to changing the register contents using a ladder diagram.

Note Input and output are actually performed when they are updated, as described in Par. 4.5.

4.6.1 Solving a Ladder Circuit (Cont'd)

As the GL40S solves a ladder circuit by scanning, the ON/OFF status of coils, latched coils or link coils, and the contents of the holding registers or the link registers may change and the new status is referred to, not only in the subsequent networks, but in the same network for the elements which are to be solved later.

In Fig. 4.7, the status of a coil solved in network N does not change between network N + 1 and the final network nor between network 1 and network N-1. But the contents of a hoding register or a link register may be changed at any time during a scanning cycle and the revised contents remain held until they are changed again.

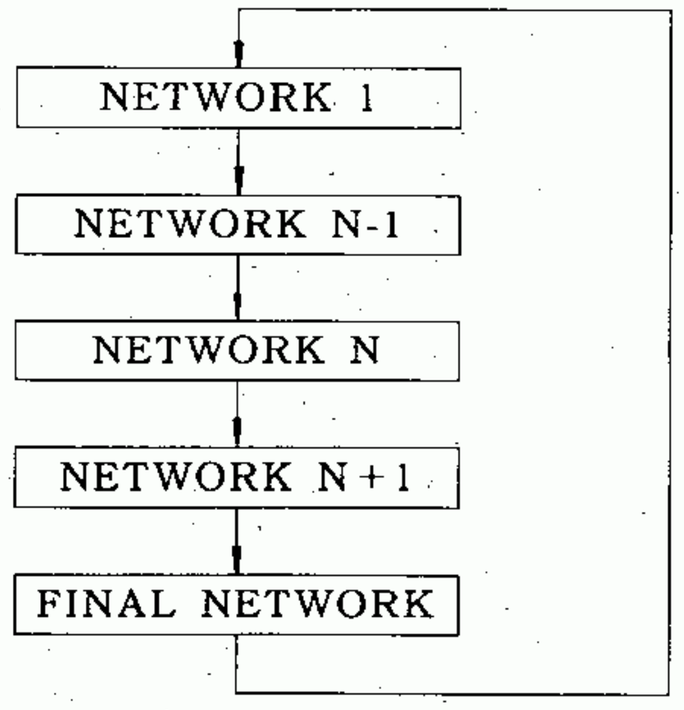


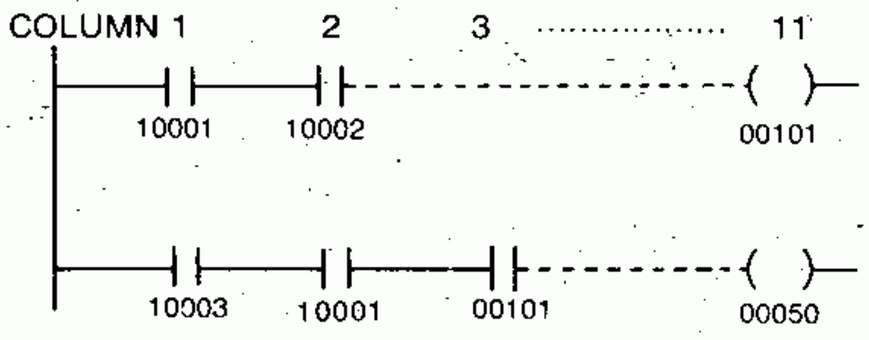
Fig. 4.7 Example of Sequence of Solving Networks

For example, a network is composed as shown in Fig. 4.8 (a). This ladder diagram is held in the memory as shown in Fig. 4.8 (b). The GL40S solves the network as shown below where the elements are identified by the reference numbers.

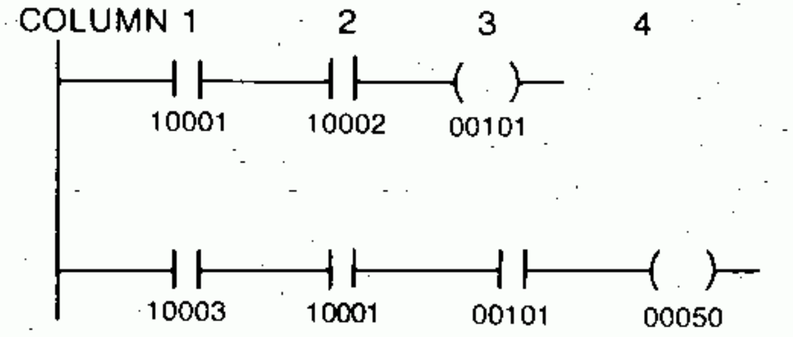
10001→10003→10002→10004→00101 (coil)→00101 (contact)→00050 (coil).

In this example, a change in coil 00101 will be reflected immediately to line 2.

Note If an element has been inserted between the contact 10002 and the coil 00101 on line 1, the change in coil 00101 will be reflected to line 2 during the next scanning cycle but not during the same scanning cycle as that of the change. If such a delay is a problem, remove line 2 from the network and place it in the next one.



(a) Display on the Screen of the Programming Panel



(b) The Network Stored in the Memory

Note Columns 4-10 are not shown above. They are indicated with dotted lines as shown above also on the programming panel.

Fig. 4.8 Example of Sequence of Solving in a Network

4.6.2 Scan Time

(1) Scan Time

The scan time T (ms) of one scanning cycle is determined roughly by the following formula.

T = (Basic Time) + (Additional Time) + (Network Processing Time) (I/O Processing Time)

Where

- Basic time = 1 ms
  - Processing time for self diagnosis
- Additional time = 3 ms
  - Processing time for communication module
- Network processing time = {(number of networks) × 3 + Σ (processing time of each element)} ÷ 1000 (ms)
  - See Table 4.5.
- I/O processing time = {(number of discrete inputs / 8 × 20) + (number of discrete outputs / 8 × 15) + (number of binary register inputs × 35) + (number of binary register outputs × 28)} ÷ 1000 (ms)

The processing time of each element varies during execution time and non-execution time as shown in Table 4.5. Therefore, the scan time usually varies. The GL40S is provided with a "constant sweep" function which keeps the scan time fixed. This function uses two hoding registers as follows.

- **Holding register 42047: stores the preset value of constant sweep given in the unit of 10ms between 10 and 200ms.**
- **Holding register 42048: stores the actual scan time when constant sweep is executed. The value varies in the unit of 10ms.**

Note If the preset value is smaller than the actual scan time, the actual time overrides the preset value.

Table 4.5 Processing Time of Elements

Element (Function)	Condition	Processing Time (μs)		Remarks
		Non Execution	Execution	
Coil, Latched Coil	—	—	1.5	
Contact, Horizontal Open/Shunt	—	—	0.125	
Transitional Contact	—	—	0.5	
Stepping Relay	—	8	8	
Timer	—	21	24	
Counter	—	21	37	
Addition	—	25	34	
Double-precision Addition	—	25	46	
Subtraction	—	25	34	
Double-precision Subtraction	—	25	50	
Multiply	0 × 0	25	34	
	9999 × 9999		45	
Double-precision Multiply	0 × 0	25	41	
	99999999 × 99999999		115	
Divide	Quotient overflow	25	43	
	For remainder		43 max.	
	For decimal part		49 max.	
Double-precision Divide	Quotient overflow	25	40	
	For remainder		337 max.	
	For decimal part		619 max.	
Signed Addition	—	25	37	
Signed Double-precision Addition	(−0) + (+0)	25	53	
	(−9999) + (−9999)		51	
Signed Subtraction	—	25	38	
Signed Double-precision Subtraction	(−0) + (+0)	25	52	
	(−9999) − (−9999)		54	
Signed Multiply	(−0) × (−0)	25	34	
	(−9999) × (−9999)		50	
Signed Divide	Quotient overflow	25	37	
	For remainder		46 max.	
	For decimal part		52 max.	
Square Root	$\sqrt{0}$	21	24	
	$\sqrt{9999}$		203	
Double-precision Square Root	$\sqrt{0}$	21	25	
	$\sqrt{99999999}$		321	
Sine	—	21	279	
Consine	—	21	284	
R-to-T Move	—	25	37	
T-to-R Move	—	25	40	
T-to-T Move	—	25	38	
FIN	—	25	37 + 1.73 × n	(1 ≦ n ≦ 100)
FOUT	Coil as destination	25	40	—
	Register as destination		38	
BLKM	Coil as destination	25	31 + 3.5 × n	(1 ≦ n ≦ 100)
	Register as destination		33 + 1.75 × n	

Table 4.5 Processing Time of Elements (Cont'd)

Element (Function)	Condition	Processing Time (μs)		Remarks
		Non Execution	Execution	
STAT	Coil as destination	21	26 + 2.25 × n	(1 ≦ n ≦ 100)
	Register as destination		24 + 0.5 × n	
SRCH	Compare	25	34 + 2 × n	(1 ≦ n ≦ 100)
	Non compare			
TSET	—	25	27 + 1.25 × n	(1 ≦ n ≦ 100)
DIBT	—	25	35 + 2 × n	(1 ≦ n ≦ 100)
DIBR	—	25	37 + 1.75 × n	
SIBT	—	25	36 + 2 × n	
SIBR	—	25	33 + 1.75 × n	
BCD → BIN	—	25	31 + 22.75 × n	(1 ≦ n ≦ 100)
BIN → BCD	—	25	31 + 16 × n	
SWAP	—	25	27 + 2.25 × n	(1 ≦ n ≦ 100)
SORT	—	25	35 + 19 × (n−1)	Max. value depending on data
BYSL	—	25	29 + 3.75 × n	(1 ≦ n ≦ 100)
BYCM	—	25	27 + 3.25 × n	(1 ≦ n ≦ 100)
BADD	Word added	25	34 + 1.25 × n	(1 ≦ n ≦ 100)
	Byte added		35 + 3.25 × n	
AND, OR, XOR	Coil as destination	25	29 + 2.25 × n	(1 ≦ n ≦ 100)
	Register as destination		31 + 4 × n	
COMP	Coil as destination	25	31 + 3.75 × n	(1 ≦ n ≦ 100)
	Register as destination		29 + 2 × n	
CMPR	Miscompare	25	2.75 × n + 5.25 × m + 40	m: Bit No. in miscompare (0 ≦ m ≦ 15)
	Non miscompare		2.75 × n + 40	
MBIT	Coil as destination	25	48	—
	Register as destination		43	
SENS	—	25	44	—
BROT	Coil as destination (right)	25	40 + 5.25 × n	(1 ≦ n ≦ 100)
	Coil as destination (left)		38 + 4 × n	
	Register as destination(right)		38 + 3.5 × n	
	Register as destination (left)		36 + 2.25 × n	
MROT	Shift	25	36 + 2.25 × n	(1 ≦ n ≦ 100)
	Rotate		36 + 2.25 × n	
TWST	—	25	21 + 17.5 × n	(1 ≦ n ≦ 100)
BCNT	—	25	32 + 17.75 × n	(1 ≦ n ≦ 100)

Note
1. The processing time for a vertical short is zero.
2. n shows table size.
3. The data given above simply provide you with a basis for calculating processing time. It is recommended to measure the actual processing time by the method shown in Fig. 5.27.

4.7 ALLOWABLE NUMBER OF MEMORY WORDS

The following formula gives the memory capacity in words needed for a network.

Memory Capacity (in Words) = 1 + (Number of Columns in Use) + (Total Number of Elements)

As an example, the detail of the network memory capacity is described using the ladder circuit of Fig. 4.9.

- Number of Columns in Use = 9 .... Columns 1-9

Note The coils A, B, and C are stored at the locations of *1, *2, and *3 in the memory as shown in Fig. 4.8 (b)

- Total Number of Elements = $\underset{\text{Line 1}}{7} + \underset{\text{Line 2}}{5} + \underset{\text{Line 3}}{6} + \underset{\text{Line 4}}{4} = 22$

Note On line 2, the locations of *4 and *5 (blank) and the vertical shunt do not occupy any memory. Horizontal shunts are counted as elements. As explained later, the timer and addition are counted as 2 and 3, respectively.

As a result,

- Memory Capacity = 1 + 9 + 22 = 32 (words)

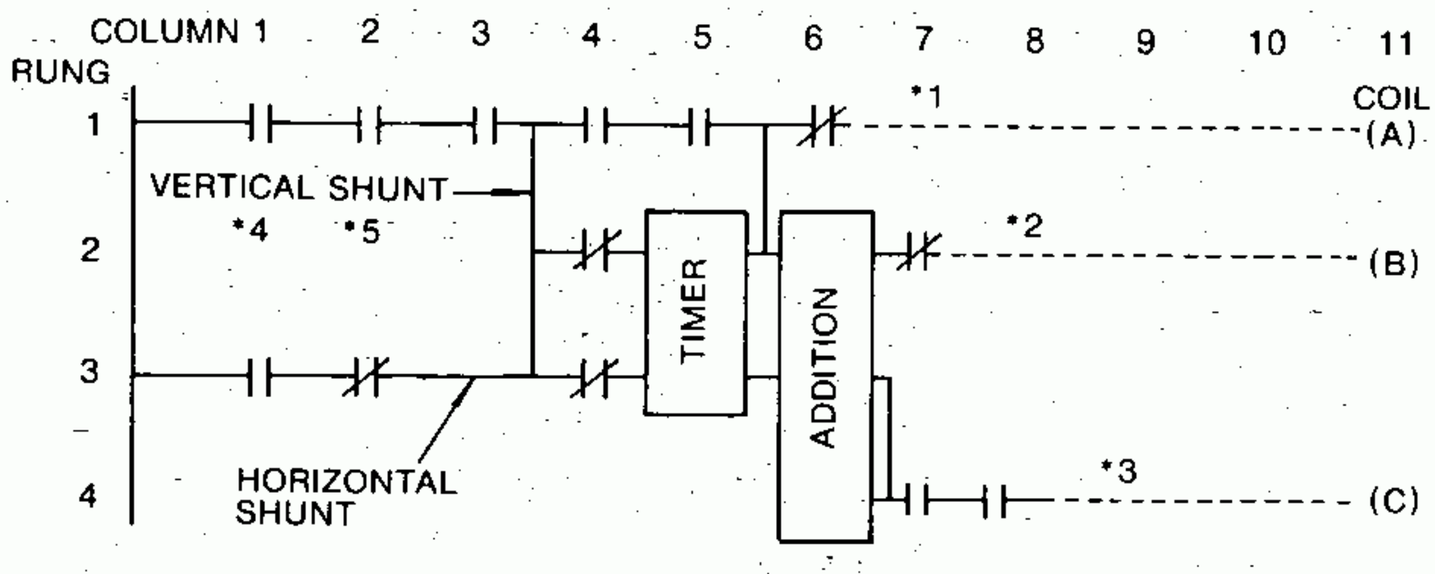
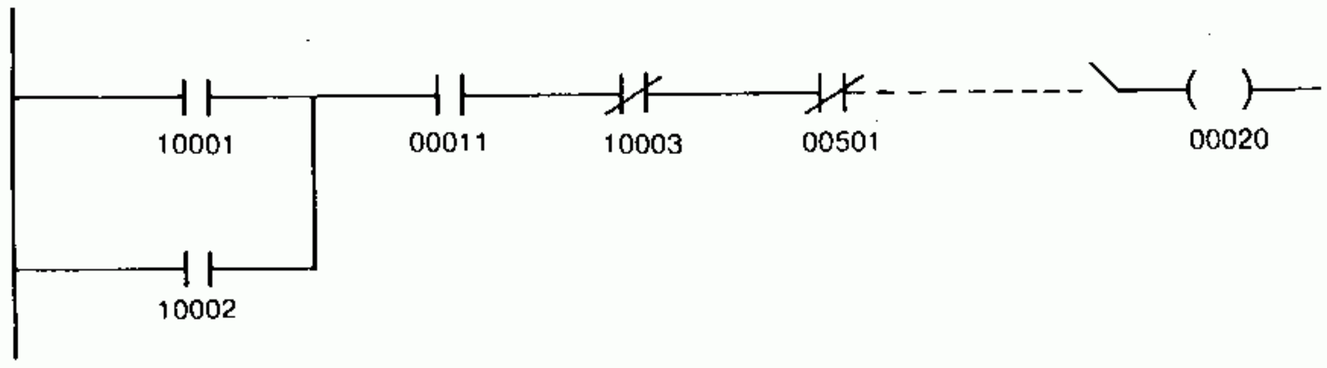


Fig. 4.9 An Example of Ladder Diagram for Calculating Memory Capacity Needed for Network

4.8 DISABLE FUNCTION

To simplify the checkout and maintenance of a control system using the GL40S Controller, a special feature is incorporated into the Controller. This feature is called the Disable funciton. The Disable status is alterable only if Memory Protect is OFF. Any logic coil selected by the cursor can be disconnected from its logic by depressing the DISABLE pushbution. If the coil was OFF when the pushbutton was depressed, it will remain OFF; if it was ON, it will remain ON. The coil is no longer controlled by the program in the Controller, but is now controlled by the operator via the P150 Programming Panel or RAP of an I/O processor module. The coil can be toggled ON/OFF/ON/OFF by successively depressing the FORCE pushbutton.

Fig. 4.10 shows an example of a disabled coil display on the screen of the programming panel.



Note The disabled condition remains unchanged even if power is turned off and on.

Fig. 4.10 Sample Display of a Disabled Coil

4.9 TRACE BACK FUNCTION

The GL40S has a trace back function in addition to a simulation function to be used when debugging is performed. The trace back function traces the ON/OFF states of the discrete signals (coils and relays) and the transition of the register values within the specified number of scan cycles.

The following parameters must be specified:

- Sampling scan cycle: The number of scans to be performed for a single sampling.
- Trigger point: The number of points to be sampled after the trigger condition is established.
- Discrete: 8 discrete points to be traced and their trigger conditions (ON/OFF or no condition).
- Register: A register to be traced and its trigger condition.

4.9 TRACE BACK FUNCTION (Cont'd)

The trace back function traces 1024 points and displays all of their states on the screen.

Fig. 4.11 is a sample of the trace back condition setting screen.

```
TRACE BACK CONDITIONS                    UNIT;001    PROGRAM MODE

   SAMPLING  ; DISABLE
SAMPLING CYCLE;    1  SCAN/POINT
   TRIGGER   ;  512 POINT

          DISCRETE                              REGISTER
 NO.      REF#          TRIGGER        REF#           TRIGGER
  1      10001            ON          40035            500
  2      00005            ON
  3      00020            *
  4       ---             --
  5       ---             --
  6       ---             --
  7       ---             --
  8       ---             --
```

Fig. 4.11 Trace Back Condition Setting

In the example above, a trigger condition is established when the scan is performed in the following conditions: the states of input relay 10001 and output relay 00005 are ON and the contents of holding register 40035 is 500. Then 512 points are traced and their states are displayed. Coil 00020 does not affect the trigger because no condition needs to be set (indicated by "*"). Thus, a trigger condition is established when all the conditions set for the specified reference are satisfied. As the trace back function displays the states of 1024 points, the states of 512 points before the establishment of the trigger condition are displayed on the same screen. If a trigger condition is established at the point before the 512th point, 0 is displayed for the value before starting tracing.

For details on the setting and the display of the trace back function, refer to P150 Programming Panel User's Manual Basic Information (SIE-C815-15.2).

SECTION 5
PROGRAMMING FUNCTIONS

5.1 PROGRAMMING FUNCTION LIST

GL40S is provided with the programming functions as shown in Table 5.1.

Table 5.1 Programming Function List

<table>
<tr><th>No.</th><th>Name</th><th>Description</th><th>Page</th></tr>
<tr><td>1</td><td>Relay</td><td>• For programming a relay circuit using relay elements.
• There are 10 types of relay elements as follows:
<table>
<tr><th>No.</th><th>Type</th><th>Symbol</th><th>No.</th><th>Type</th><th>Symbol</th></tr>
<tr><td>1</td><td>Normally Open Contact</td><td>⊣ ⊢</td><td>6</td><td>Vertical Shunt</td><td>|</td></tr>
<tr><td>2</td><td>Normally Closed Contact</td><td>⊣/⊢</td><td>7</td><td>Vertical Open</td><td>Non</td></tr>
<tr><td>3</td><td>Transitional Contact (OFF to ON)</td><td>⊣↑⊢</td><td>8</td><td>Coil</td><td>⊣( )⊢</td></tr>
<tr><td>4</td><td>Transitional Contact (ON to OFF)</td><td>⊣↓⊢</td><td>9</td><td>Latched Coil</td><td>⊣(L)⊢</td></tr>
<tr><td>5</td><td>Horizontal Shunt</td><td>—</td><td></td><td></td><td></td></tr>
</table></td><td>46</td></tr>
<tr><td>2</td><td>Timer</td><td>• For counting a time while any signal is ON.
• There are 3 types of timers as follows:
<table>
<tr><th>No.</th><th>Type</th><th>Symbol</th></tr>
<tr><td>1</td><td>Timer (Seconds)</td><td>T1.0</td></tr>
<tr><td>2</td><td>Timer (Tenths of Seconds)</td><td>T0.1</td></tr>
<tr><td>3</td><td>Timer (Hundredths of Seconds)</td><td>T.01</td></tr>
</table></td><td>63</td></tr>
<tr><td>3</td><td>Counter</td><td>• For counting the number when any signal is changed from OFF to ON.
• There are 2 types of counters as follows:
<table>
<tr><th>No.</th><th>Type</th><th>Symbol</th></tr>
<tr><td>1</td><td>Up Counter</td><td>UCTR</td></tr>
<tr><td>2</td><td>Down Counter</td><td>DCTR</td></tr>
</table></td><td>70</td></tr>
<tr><td>4</td><td>Arith-metic</td><td>• Arithmetic operation for numerical values in 4-digit decimal form or 8-digit.
• There are 8 types of arithmetics as follows:
<table>
<tr><th>No.</th><th>Type</th><th>Symbol</th><th>Functions</th></tr>
<tr><td>1</td><td>Addition</td><td>ADD</td><td>Addition in 4-digit decimal</td></tr>
<tr><td>2</td><td>Subtraction</td><td>SUB</td><td>Subtraction in 4-digit decimal</td></tr>
<tr><td>3</td><td>Multiply</td><td>MUL</td><td>Multiply in 4-digit decimal</td></tr>
<tr><td>4</td><td>Divide</td><td>DIV</td><td>Divide in 4-digit decimal</td></tr>
<tr><td>5</td><td>Double-precision Addition</td><td>DADD</td><td>Addition in 8-digit decimal</td></tr>
<tr><td>6</td><td>Double-precision Subtraction</td><td>DSUB</td><td>Subtraction in 8-digit decimal</td></tr>
<tr><td>7</td><td>Double-precision Multiply</td><td>DMUL</td><td>Multiply in 8-digit decimal</td></tr>
<tr><td>8</td><td>Double-precision Divide</td><td>DDIV</td><td>Divide in 8-digit decimal</td></tr>
</table></td><td>80</td></tr>
<tr><td>5</td><td>Signed Arith-metic</td><td>• Signed addition/subtraction operation for numerical values in 4-digit decimal form or 8-digit.
• Signed multiply/divide operation for numerical values in 4-digit decimal form.
• There are 6 types of signed arithmetics as follows:
<table>
<tr><th>No.</th><th>Type</th><th>Symbol</th><th>Functions</th></tr>
<tr><td>1</td><td>Signed Addition</td><td>SADD</td><td>Signed Addition in 4-digit decimal</td></tr>
<tr><td>2</td><td>Signed Subtraction</td><td>SSUB</td><td>Signed Subtraction in 4-digit decimal</td></tr>
<tr><td>3</td><td>Signed Multiply</td><td>SMUL</td><td>Signed Multiply in 4-digit decimal</td></tr>
<tr><td>4</td><td>Signed Divide</td><td>SDIV</td><td>Signed Divide in 4-digit decimal</td></tr>
<tr><td>5</td><td>Signed Double-precision Addition</td><td>SDAD</td><td>Signed Addition in 8-digit decimal</td></tr>
<tr><td>6</td><td>Signed Double-precision Subtraction</td><td>SDSB</td><td>Signed Subtraction in 8-digit decimal</td></tr>
</table></td><td>105</td></tr>
</table>

Table 5.1 Programming Function List (Cont'd)

<table>
<tr><th>No.</th><th>Name</th><th>Description</th><th>Page</th></tr>
<tr><td>6</td><td>Square Root</td><td>• Square root operation for numerical values in 4-decimal form or 8-digit.
• There are two types of square root as follows:
<table><tr><th>No.</th><th>Type</th><th>Symbol</th><th>Functions</th></tr><tr><td>1</td><td>Square Root</td><td>SQRT</td><td>In 4-digit decimal</td></tr><tr><td>2</td><td>Double-precision Square Root</td><td>DSQR</td><td>In 8-digit decimal</td></tr></table></td><td>123</td></tr>
<tr><td>7</td><td>Trigono-metric Function</td><td>• Trigonometric function operation for numeric data of up to four decimal places within 360 degrees.
• There are two types of trigonometric function operations.
<table><tr><th>No.</th><th>Type</th><th>Symbol</th><th>Functions</th></tr><tr><td>1</td><td>Sine</td><td>SIN</td><td>Sine operation for numeric data within 360°</td></tr><tr><td>2</td><td>Cosine</td><td>COS</td><td>Cosine operation for numeric data within 360°</td></tr></table></td><td>128</td></tr>
<tr><td>8</td><td>Move</td><td>• A variety types of data move in multi-data groups.
• There are 9 types of data move as follows:
<table><tr><th>No.</th><th>Type</th><th>Symbol</th></tr><tr><td>1</td><td>Block Move</td><td>BLKM</td></tr><tr><td>2</td><td>Register-to-Table</td><td>R → T</td></tr><tr><td>3</td><td>Table-to-Register</td><td>T → R</td></tr><tr><td>4</td><td>Table-to-Table</td><td>T → T</td></tr><tr><td>5</td><td>First In</td><td>FIN</td></tr><tr><td>6</td><td>First Out</td><td>FOUT</td></tr><tr><td>7</td><td>Table Search</td><td>SRCH</td></tr><tr><td>8</td><td>Table Set</td><td>TSET</td></tr><tr><td>9</td><td>Get Controller System Status</td><td>STAT</td></tr></table></td><td>133</td></tr>
<tr><td>9</td><td>Block Move with Index</td><td>• For specifying a data group transfered by an index, or a destination.
• There are 4 types of block move with index as follows:
<table><tr><th>No.</th><th>Type</th><th>Symbol</th></tr><tr><td>1</td><td>Block Move 1 with Destination Index</td><td>DIBT</td></tr><tr><td>2</td><td>Block Move 2 with Destination Index</td><td>DIBR</td></tr><tr><td>3</td><td>Block Move 1 with Source Index</td><td>SIBT</td></tr><tr><td>4</td><td>Block Move 2 with Source Index</td><td>SIBR</td></tr></table></td><td>166</td></tr>
<tr><td>10</td><td>Data Conversion</td><td>• For changing data pattern within the matrix.
• There are 7 types of data conversion.
<table><tr><th>No.</th><th>Type</th><th>Symbol</th></tr><tr><td>1</td><td>BCD → BIN Conversion</td><td>BIN</td></tr><tr><td>2</td><td>BIN → BCD Conversion</td><td>BCD</td></tr><tr><td>3</td><td>Swap</td><td>SWAP</td></tr><tr><td>4</td><td>Sort</td><td>SORT</td></tr><tr><td>5</td><td>Byte Split</td><td>BYSL</td></tr><tr><td>6</td><td>Byte Composition</td><td>BYCM</td></tr><tr><td>7</td><td>Block Addition</td><td>BADD</td></tr></table></td><td>180</td></tr>
</table>

Table 5.1 Programming Function List (Cont'd)

<table>
<tr><th>No.</th><th>Name</th><th>Description</th><th>Page</th></tr>
<tr><td>11</td><td>Matrix</td><td>• For operating upon bit patterns within the matrix.
• There are 11 types of matrices as follows:
<table><tr><th>No.</th><th>Type</th><th>Symbol</th></tr><tr><td>1</td><td>Logical AND</td><td>AND</td></tr><tr><td>2</td><td>Logical OR</td><td>OR</td></tr><tr><td>3</td><td>Logical Exclusive OR</td><td>XOR</td></tr><tr><td>4</td><td>Logical Complement</td><td>COMP</td></tr><tr><td>5</td><td>Logical Compare</td><td>CMPR</td></tr><tr><td>6</td><td>Logical Bit Modify</td><td>MBIT</td></tr><tr><td>7</td><td>Logical Bit Sense</td><td>SENS</td></tr><tr><td>8</td><td>Logical Bit Rotate</td><td>BROT</td></tr><tr><td>9</td><td>Logical Multi-Bit Rotate</td><td>MROT</td></tr><tr><td>10</td><td>Logical Byte Rearrangement</td><td>TWST</td></tr><tr><td>11</td><td>Logical Bit Count</td><td>BCNT</td></tr></table></td><td>197</td></tr>
<tr><td>12</td><td>Skip</td><td>• Any networks are skipped and not solved.
• Symbol: SKP</td><td>224</td></tr>
<tr><td>13</td><td>Stepping Switch</td><td>• There are 32 stepping switches. 2YYXX (YY: 01-32, XX: 01-99)
• Each stepping switch can have max. 99 step independently.</td><td>225</td></tr>
<tr><td>14</td><td>Subroutine</td><td>• This function calls up subroutines.
• A symbol consisting of a two-digit subroutine number preceded by G is used to call up a subroutine.</td><td>228</td></tr>
<tr><td>15</td><td>Communi-cation</td><td>• Communication can be performed via communication port COMM (the GL40S can be used as a master).
• COMM is used as a communication symbol.</td><td>230</td></tr>
<tr><td>16</td><td>Motion Control</td><td>• Motion control can be performed by using servo interface module (IF66).
• There are 15 kinds of commands.
<table><tr><th>No.</th><th>Name</th><th>Symbol</th></tr><tr><td>1</td><td>1-axis program operation</td><td>MOVI</td></tr><tr><td>2</td><td>1-axis single block operation</td><td>MVIS</td></tr><tr><td>3</td><td>Independent multi-axis program operation</td><td>MOVM</td></tr><tr><td>4</td><td>3-axis program operation</td><td>MOVL</td></tr><tr><td>5</td><td>3-axis single block operation</td><td>MVLS</td></tr><tr><td>6</td><td>1-axis zero point return operation</td><td>ZRN</td></tr><tr><td>7</td><td>JOG operation</td><td>JOG</td></tr><tr><td>8</td><td>Handle operation</td><td>HNDL</td></tr><tr><td>9</td><td>Monitor</td><td>MON</td></tr><tr><td>10</td><td>Current position setting</td><td>POS</td></tr><tr><td>11</td><td>Parameter setting</td><td>PRM</td></tr><tr><td>12</td><td>Variable setting</td><td>VAR</td></tr><tr><td>13</td><td>Alarm reset</td><td>ARES</td></tr><tr><td>14</td><td>Servo ON</td><td>SVON</td></tr><tr><td>15</td><td>Current position adjustment</td><td>ADJ</td></tr></table></td><td>254</td></tr>
</table>

5.2 RELAYS

5.2.1 Relay Logic Elements

Table 5.2 shows relay elements for programming a relay circuit.

Table 5.2 Relay Logic Elements

Element		Symbol (×××××: Reference Number)	Description	Reference Number
Contact	Normally Open	⊣ ⊢ ×××××	Operates while reference coil* is ON.	Coil: 00001 to 02048† Input relay: 10001 to 10512 Stepping relay: 2YYXX (YY: 01 to 32 XX: 01 to 99) Link coil: D0001 to D1024
	Normally Closed	⊣/⊢ ×××××	Operates while reference coil is OFF.	
	Transitional Contact (OFF to ON)	⊣↑⊢ ×××××	Operates only in 1 scan cycle when reference coil is turned on.	
	Transitional Contact (ON to OFF)	⊣↓⊢ ×××××	Operates only in 1 scan cycle when reference coil is turned off.	
Connection	Horizontal Shunt	—	Shunts between columns.	—
	Vertical Shunt	|	Shunts between lines.	
	Vertical Open	⋮	Opens the vertical shunt.	
Coil	Coil	−( )− ×××××	De-energized when power is restored.	Coil: 00001 to 02047 Link coil: D0001 to D1024
	Latched Coil	−(L)− ×××××	Energized coil when power is restored.	

* Meaning of reference coil:

- Reference coil of $\underset{0\times\times\times\times}{\dashv\vdash}$ is $\underset{0\times\times\times\times}{-(\ )-}$
- Reference coil of $\underset{1\times\times\times\times}{\dashv\vdash}$ is input relay $1\times\times\times\times$.

† Coil 02048 cannot be programmed, because it is used as a battery monitoring coil. But its contact can be used in any network.

(1) NO, NC Contacts

① Symbol:

Note × × × × × Shows reference No.

② Function

NO contact: Operates while reference coil is on, and signals are sent from left to right.

NC contact: Operates while reference coil is off, and signals are sent from right to left.

Note **Reference coils are coil and relay which make contact conduct or not conduct.**

③ Reference No.

Table 5.3 NO, NC Contacts Reference No.

Reference Coil	Reference No.
Input relay	10001 to 10512
Coil, latched coil	00001 to 02048*
Stepping relay	2YYXX (YY: 01 to 32, XX: 01 to 99)
Link coil	D0001 to D1024

* 02048 is a battery coil.

④ Operation

Example:

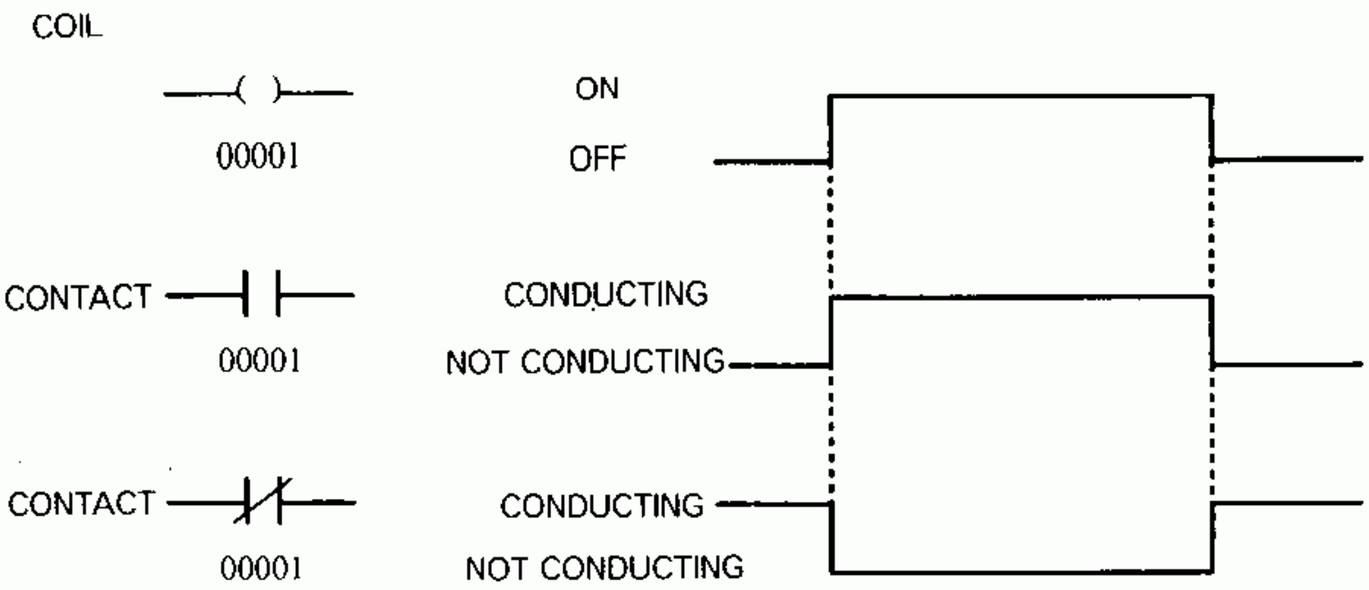


Fig. 5.1 NO, NC Contacts Operation

(2) Transitional Contacts

① Symbol:

Note × × × × × Shows reference No.

5.2.1 Relay Logic Elements (Cont'd)

② Function

Transitional contact: (OFF to ON) — Operates only in 1 scan cycle when reference coil is turned on, and signals are sent from left to right.

Transitional contact: (ON to OFF) — **Operates only is 1 scan cycle when reference coil is turned off, and signals are sent from left to right.**

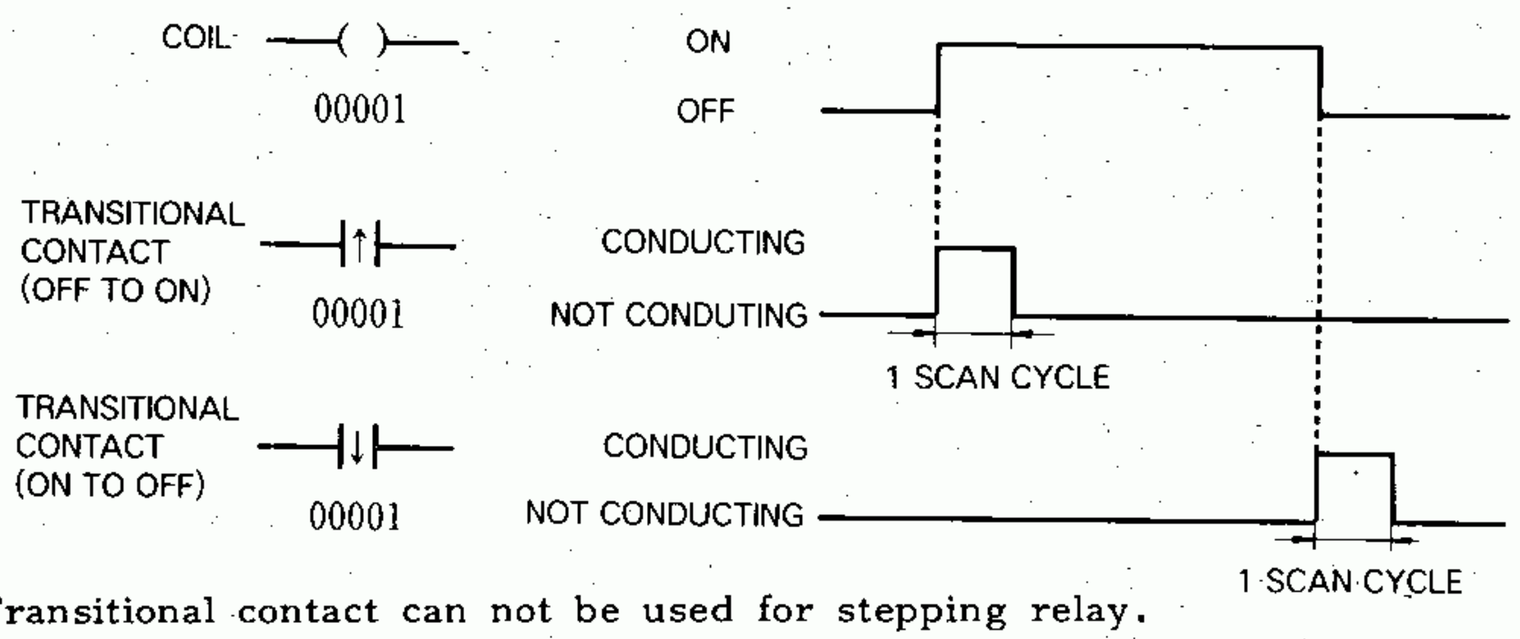
③ Reference No.

Table 5.4 Transitional Contacts Reference No.

Reference Coil	Reference No.
Input relay	10001 to 10512
Coil, latched coil	00001 to 02048
Link coil	D0001 to D1024

④ Operation

Example:



Note Transitional contact can not be used for stepping relay.

Fig. 5.2 Transitional Operation

(3) Horizontal Shunt

① Symbol: ———

② Function: **Shunts between columns, signals are sent from left to right.**

③ Reference No.: None

④ Application

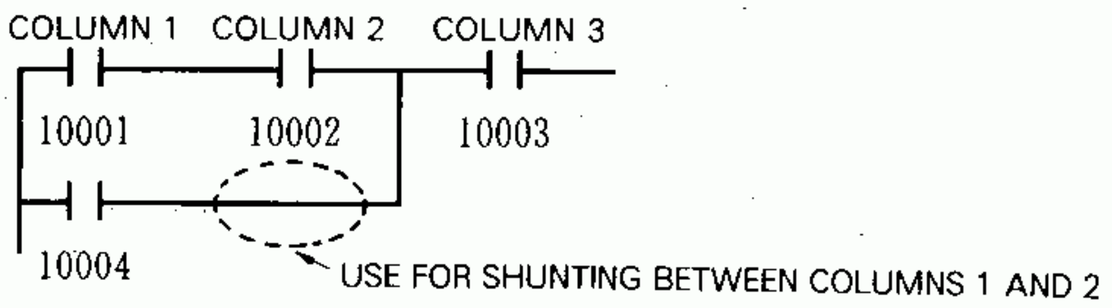


Fig. 5.3 Application for Horizontal Shunt

(4) Horizontal Open

① Symbol: • •

② Function: Opens horizontal shunt.

③ Reference No.: None

④ Application

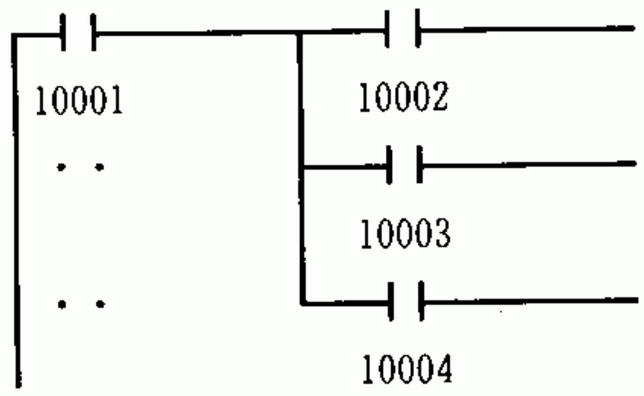


Fig. 5.4 Application for Horizontal Open

(5) Vertical Shunt

① Symbol: |

② Function: Shunts between columns, and signals are sent from top to down, down to top.

③ Reference No.: None

④ Application

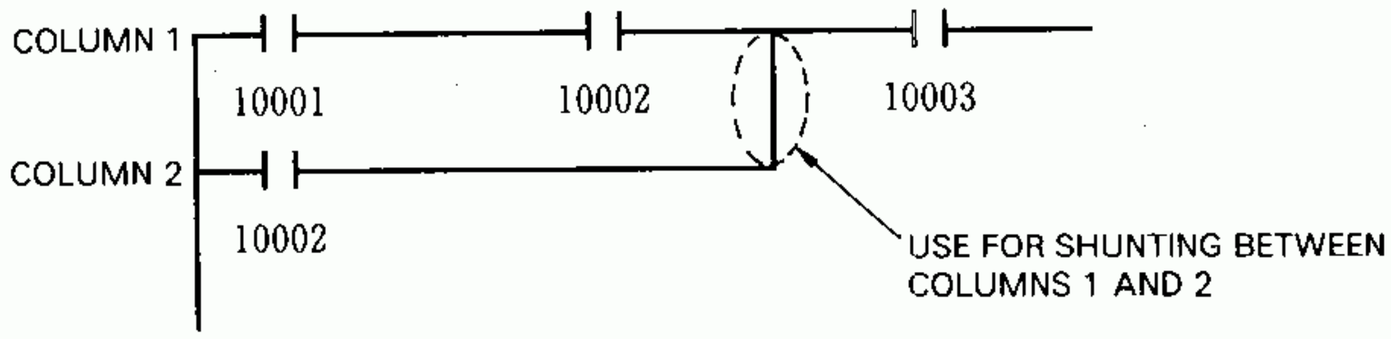


Fig. 5.5 Application for Vertical Shunt

(6) Vertical Open

① Symbol: None

② Function: Opens vertical shunt.

③ Reference No.: None

④ Application

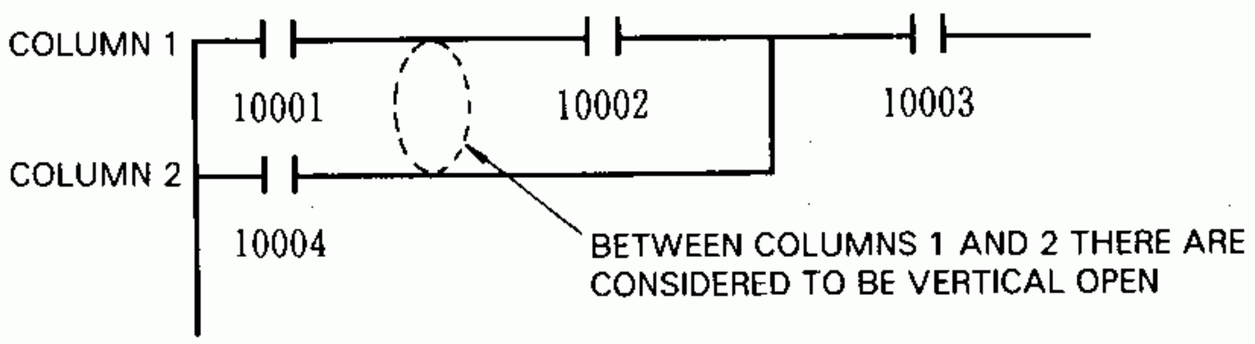


Fig. 5.6 Application for Vertical Open

(7) Coil

① Symbol: ⊣( )⊢
×××××

② Type, function, reference No.

Table 5.5 Coil Type, Function, Reference No.

Type	Function	Reference No.
Output Coil	Outputs ON/OFF condition externally through output module.	00001 to 00512
Internal Coil	Used for assembling logic. Cannot output ON/OFF condition externally.	04097 to 02047
Battery Monitor Coil	Monitors memory back-up battery output voltage.	02048

③ Output coil operation

Example:

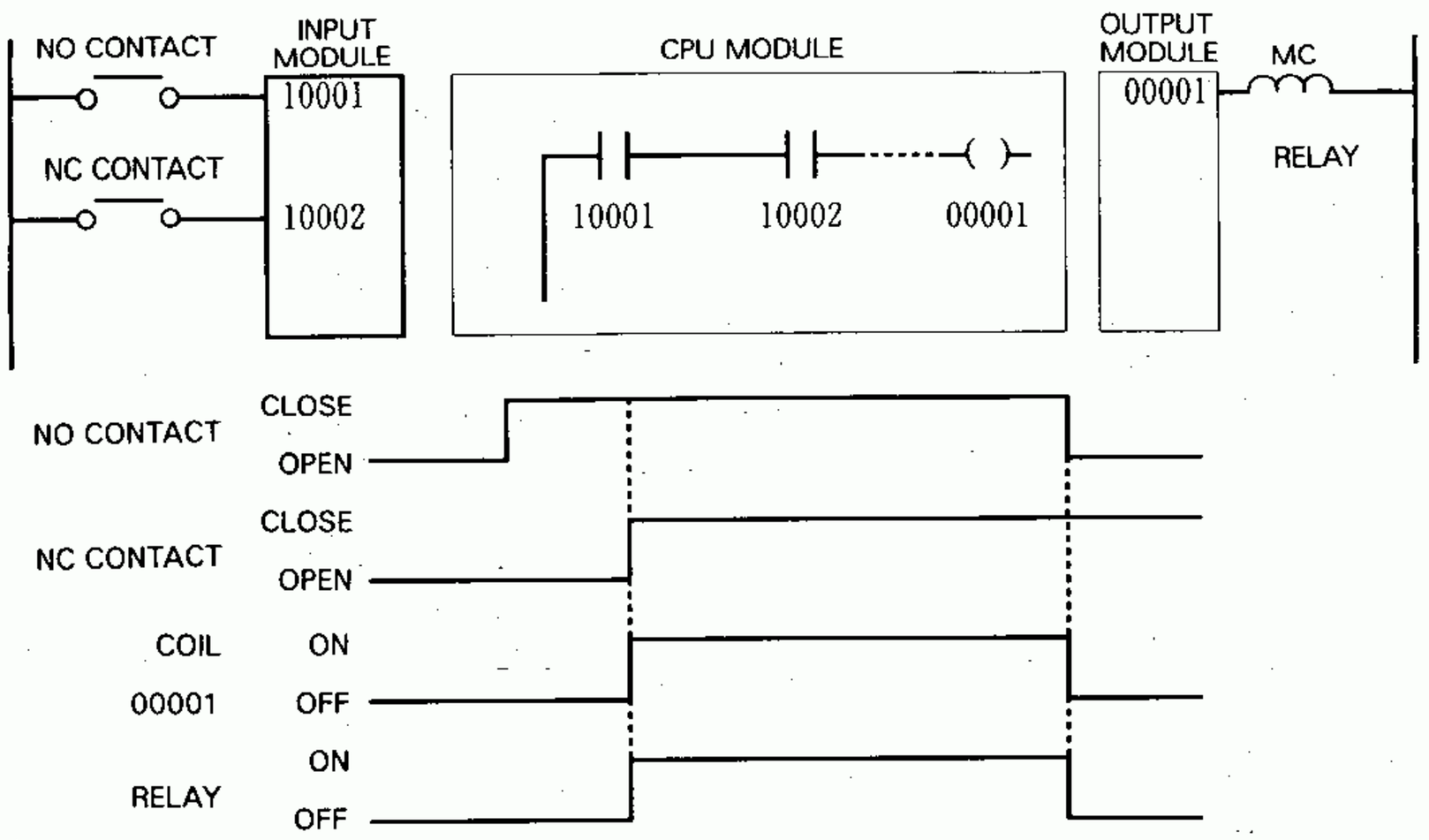


Fig. 5.7 Output Coil Operation

④ Internal coil application

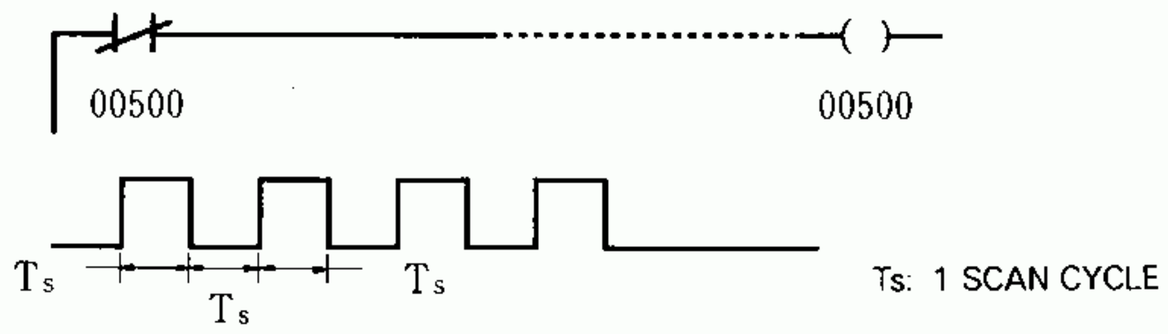


Fig. 5.8 Application for Internal Coil

⑤ Battery monitor coil operation

Example:

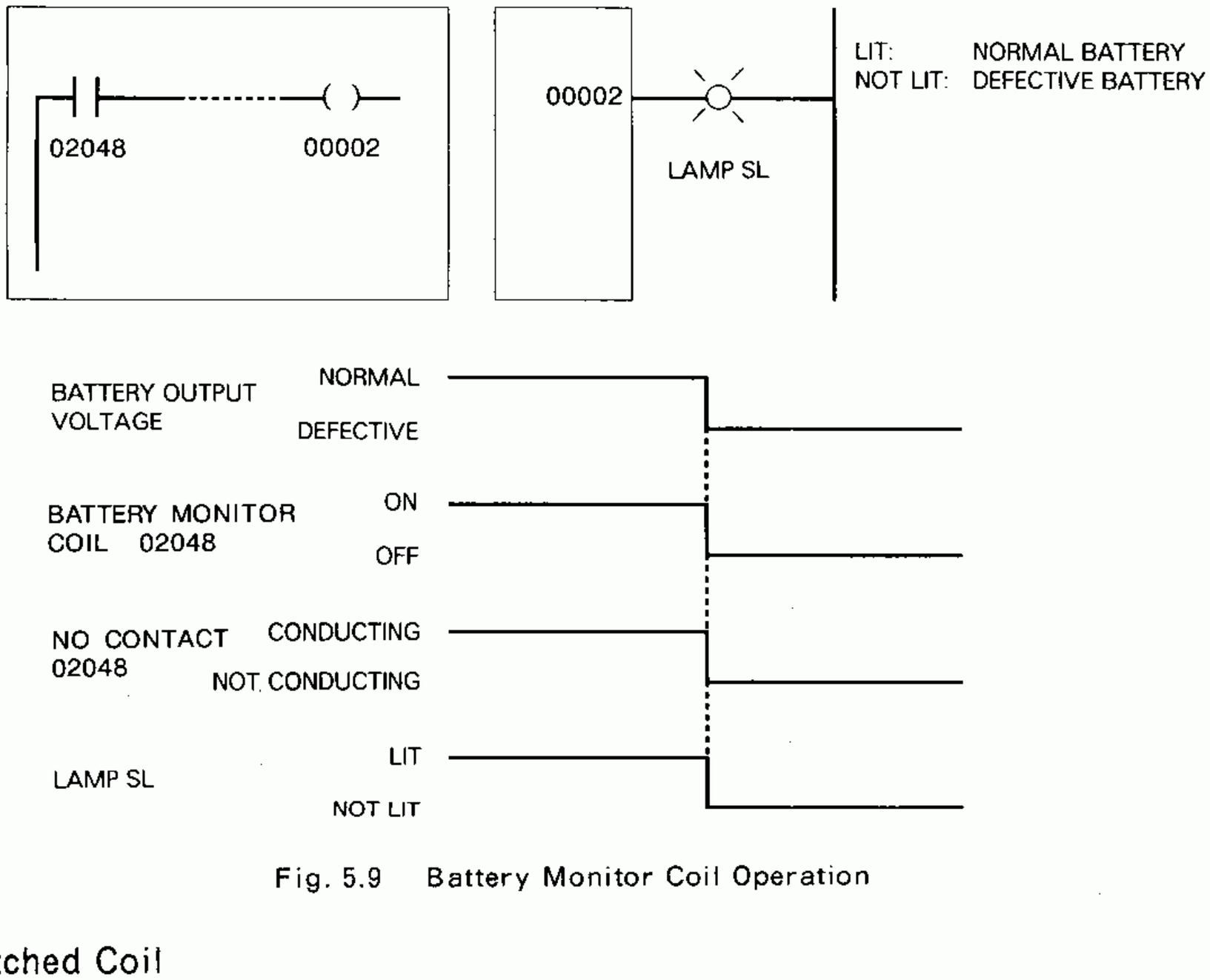


Fig. 5.9 Battery Monitor Coil Operation

(8) Latched Coil

① Symbol: $\overset{-(L)-}{\times\times\times\times\times}$

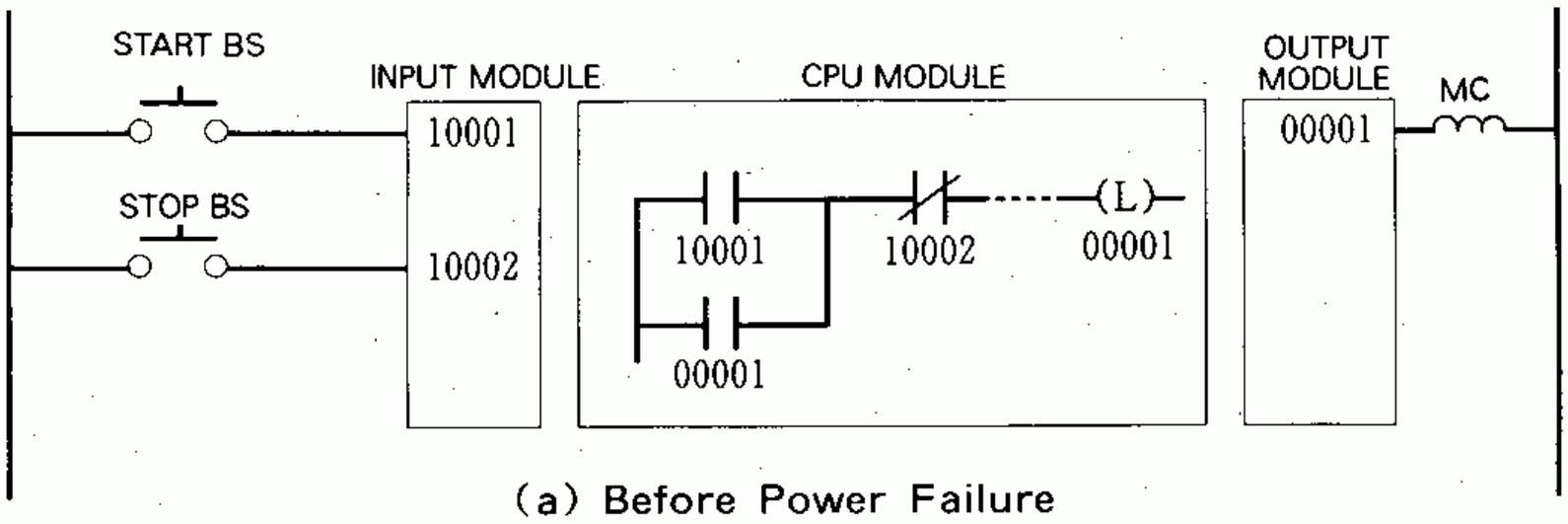
② Function: The ON/OFF status before power failure will be restored after recovery.

③ Reference No: The same as coil reference No.

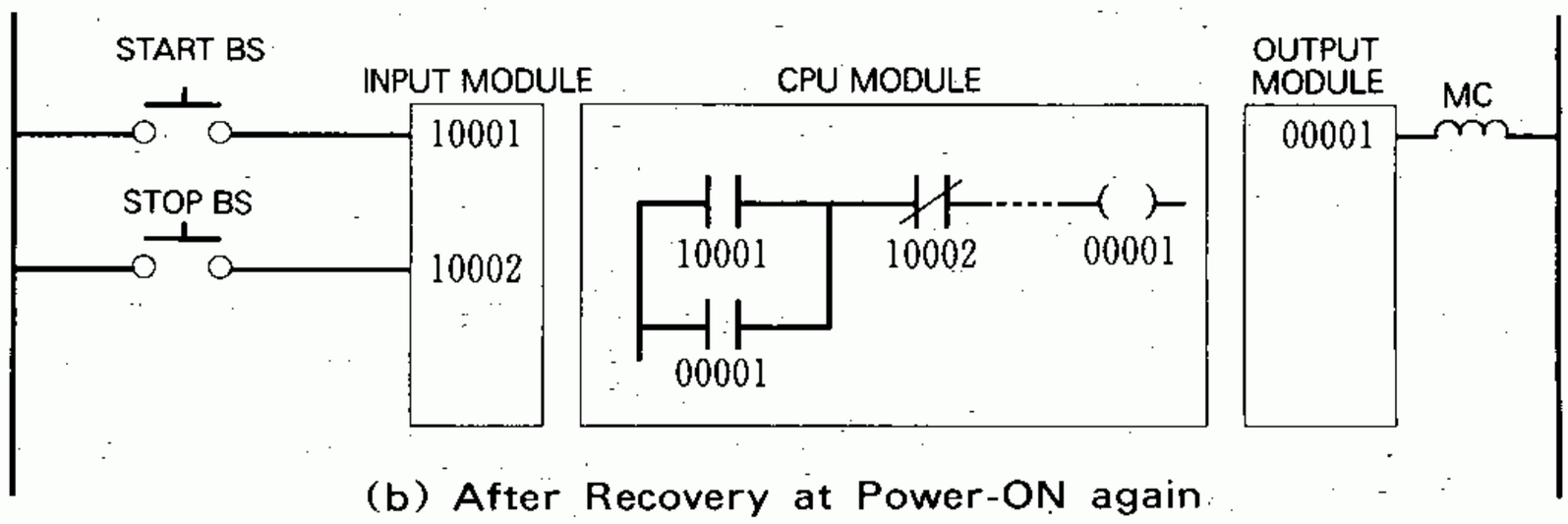
5.2.1 Relay Logic Elements (Cont'd)

④ Operation

Example:



(a) Before Power Failure



(b) After Recovery at Power-ON again

Fig. 5.10 Latched Coil Operation

When —(L)— 00001 is ON and a power failure occurs;
At recovery, without depressing start BS,
—(L)— 00001 is ON again.

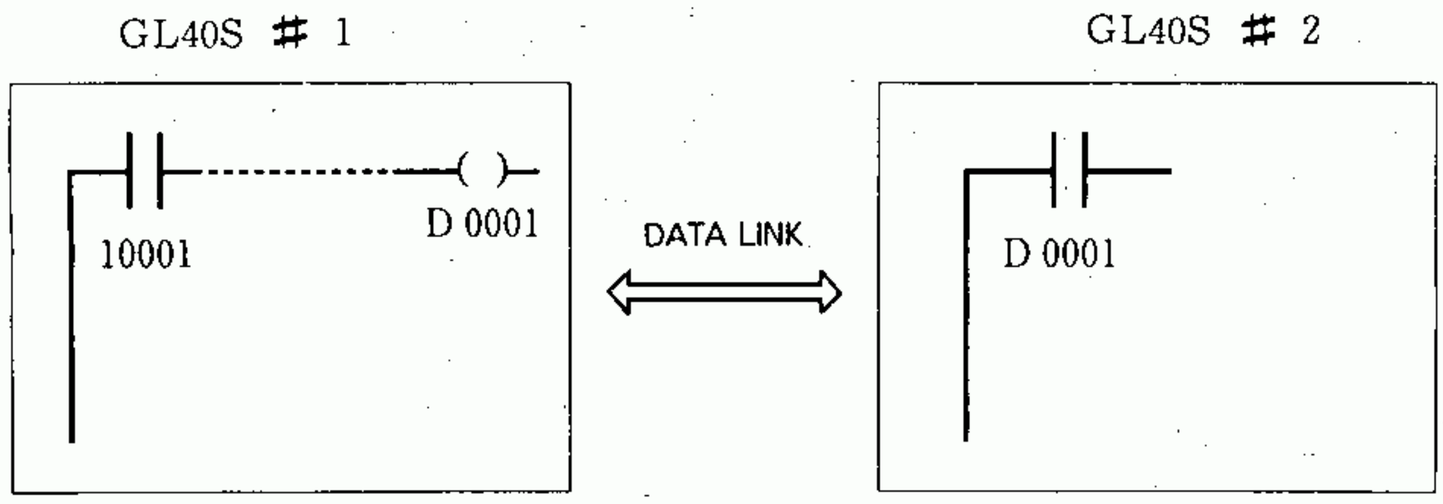
(9) Link Coil

① Symbol: $\dashv(\ )\vdash$ D××××

② Function: When some units of GL40S are provided at high speed data link, this coil makes other unit refer to them.

③ Reference No.: D0001 to D1024

④ Operation



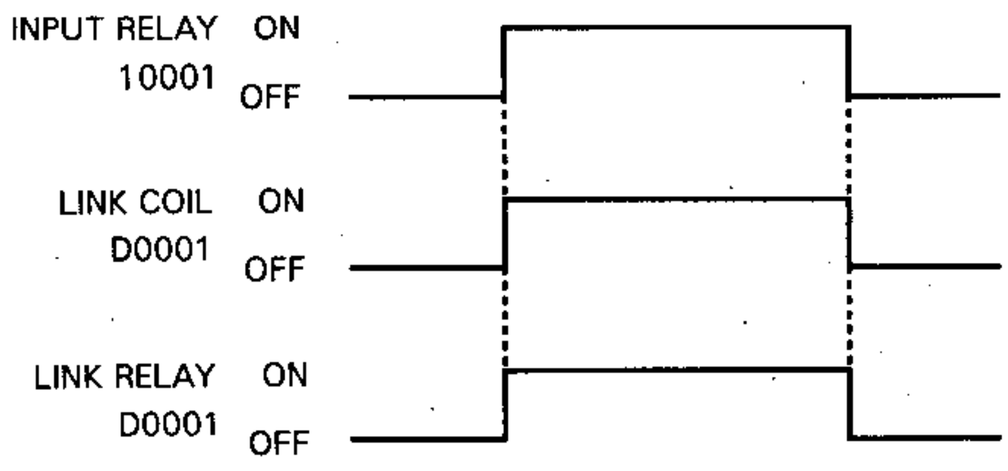
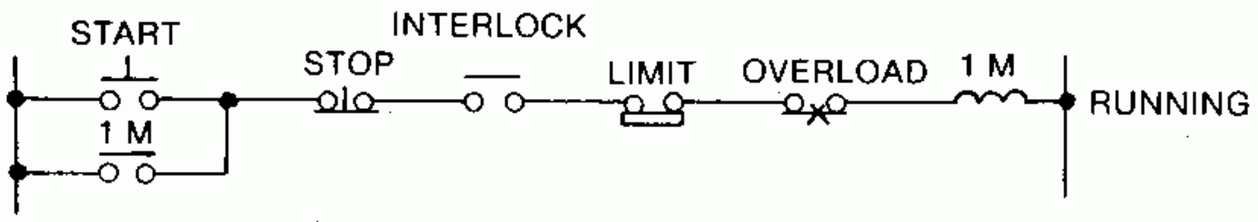


Fig. 5.11 Link Coil Operation

5.2.2 Example Relay Logic Circuit

Fig. 5.12 shows an example of relay logic circuit.

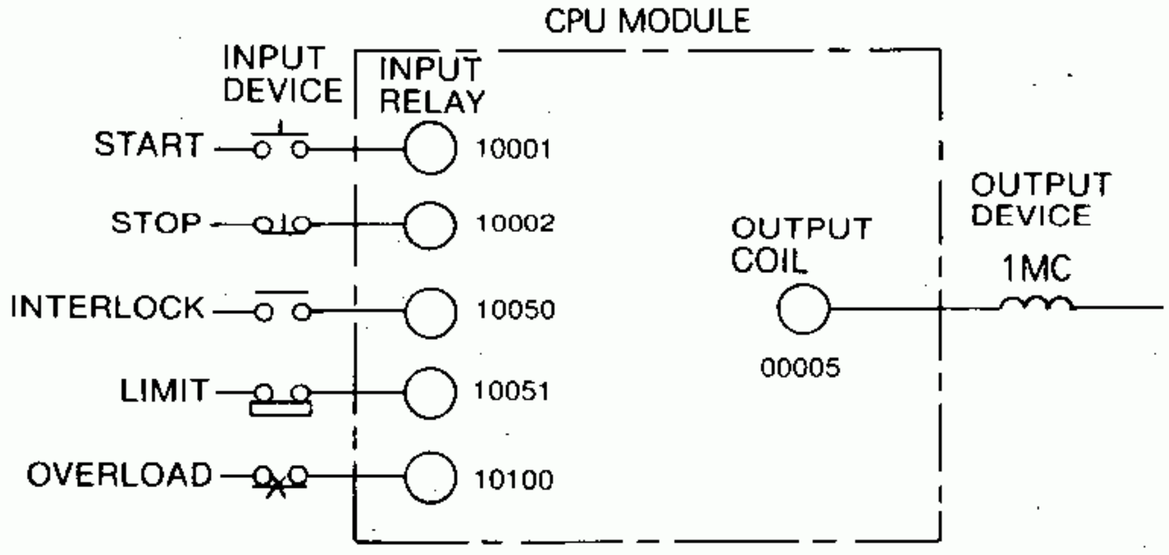
- If the logic in Fig. 5.12 (a) is to be implemented in the GL40S controller, the control elements must be connected to input circuits in the I/O configuration and outputs assigned.



(a) Example Relay Logic

⇩

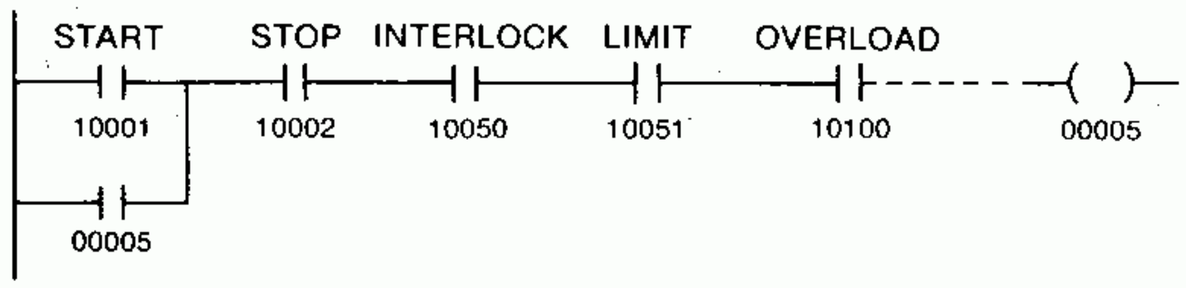
- Fig. (b) illustrates assumed input assignments and wiring details. Output number 00005 is assigned to operate the external device.



(b) Assumed I/O Wiring

⇩

- The resultant internal logic to be programmed by the user is shown in Fig. (c).



(c) Equivalent GL40S Program

Note

1. In this example, the stop, interlock, limit and overload are normally on for safety from failure. Thus they are normally open contacts in the ladder diagram. Whether a signal is a normally open (NO) or a normally closed (NC) contact should be decided in the design stage.
2. The reference number of a coil is given, like "00005," just as appearing on the screen of the programming panel. On the drawing, "–( )" may be written instead of "00005."

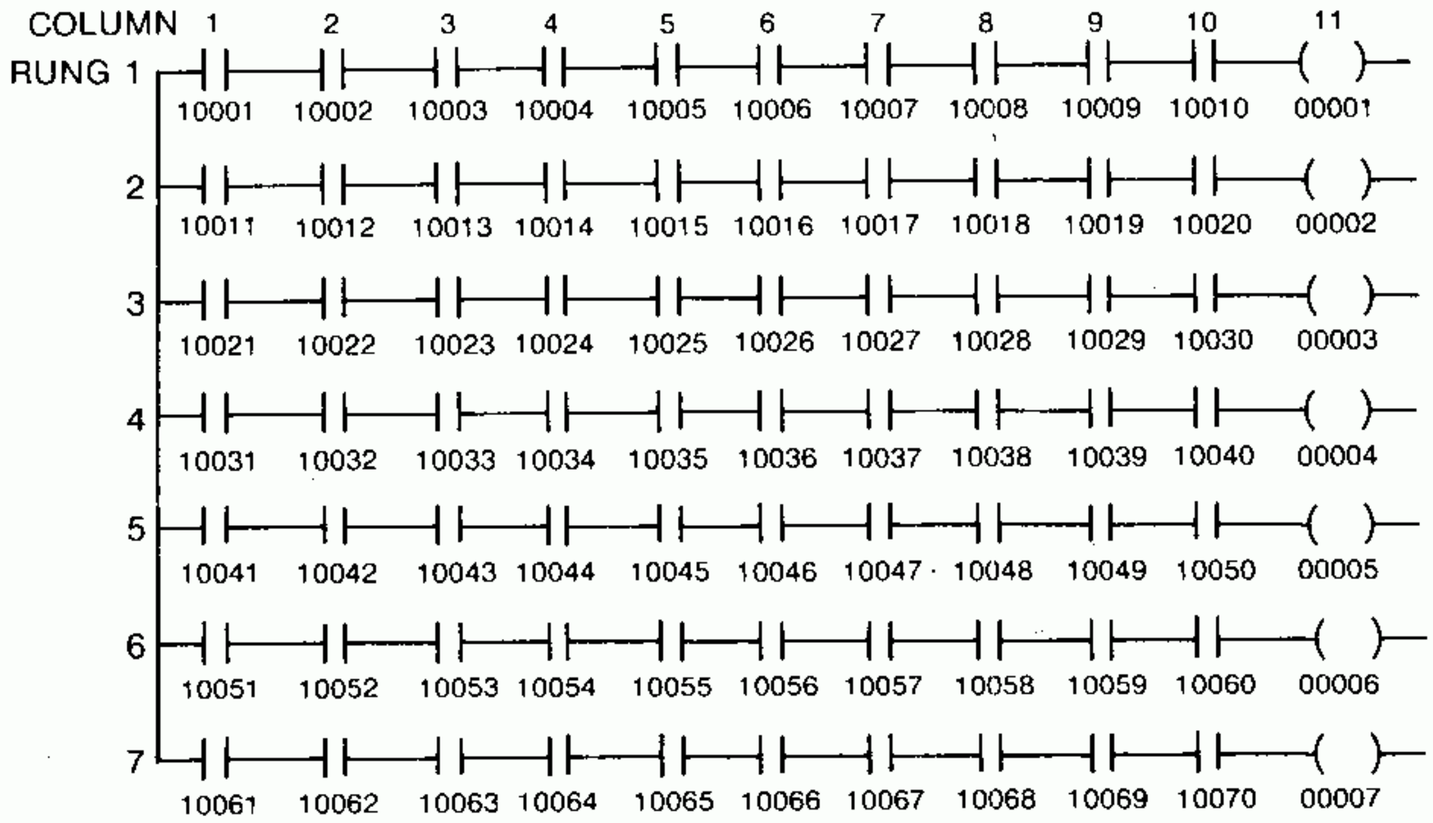
5

Fig. 5.12 Sample Relay Logic Circuit

5.2.3 Creating of Relay Circuits

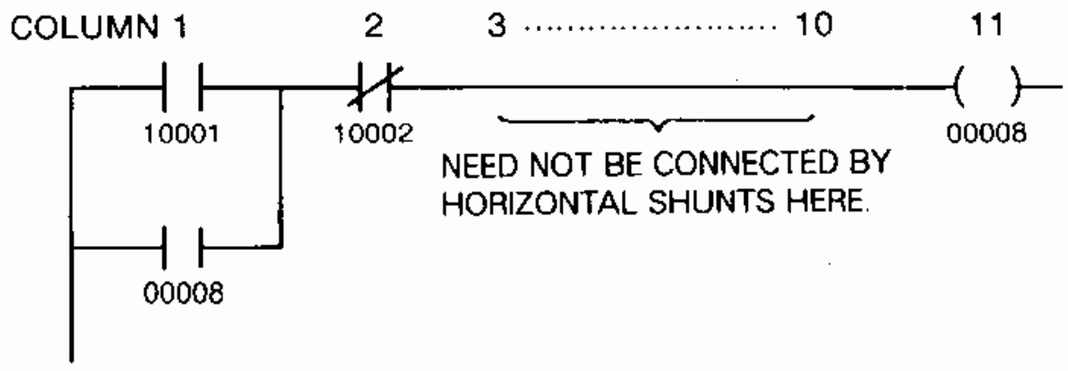
(1) In a network, contacts and horizontal short elements (shunts) may exist at any intersection of the matrix of 7 lines by 10 columns. Coils may exist on column 11. Thus up to 70 contacts and seven coils can be used in a network.

Example:



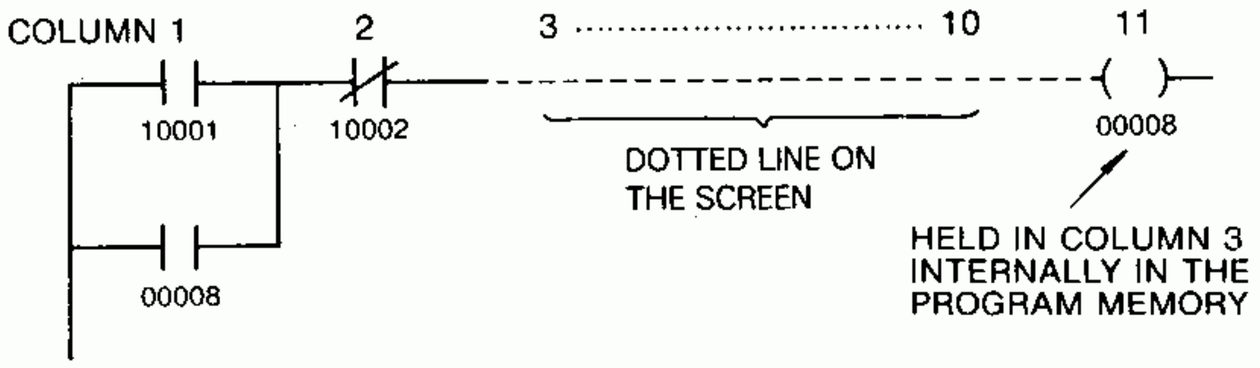
(2) If a relay circuit consisting of contacts and a uniting element ends at coulmn 2 (see below), columns 3 to 10 need not be connected by horizontal shunts before placing a coil in column 11.

Example:



By placing a coil in coumn 3 on the screen of the programming panel, it will automatically appear in coulmn 11 as shown below.

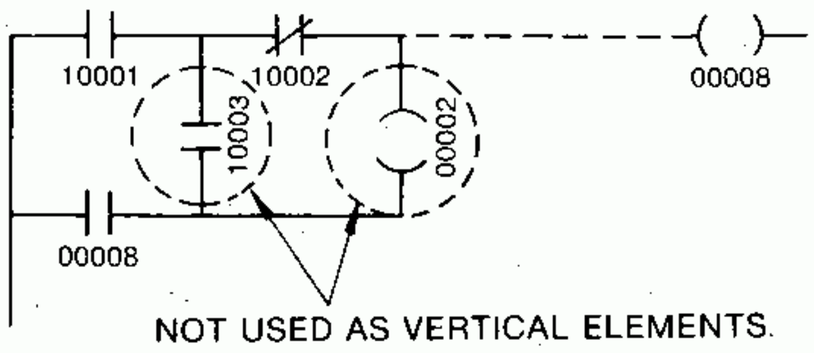
Example:



5.2.3 Creating of Relay Circuits (Cont'd)

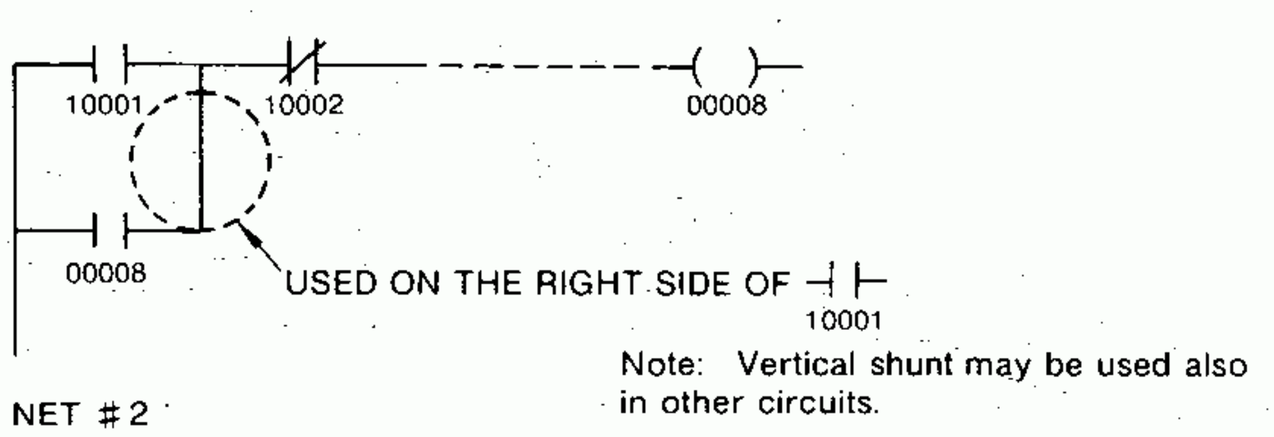
(3) Contacts and coils are used as horizontal elements, but not as vertical elements.

Example:



(4) A vertical short may exist on the right side of a contact or horizontal shunt element for downward connection (to the next line), but not on line 7 or with any coil.

Example:

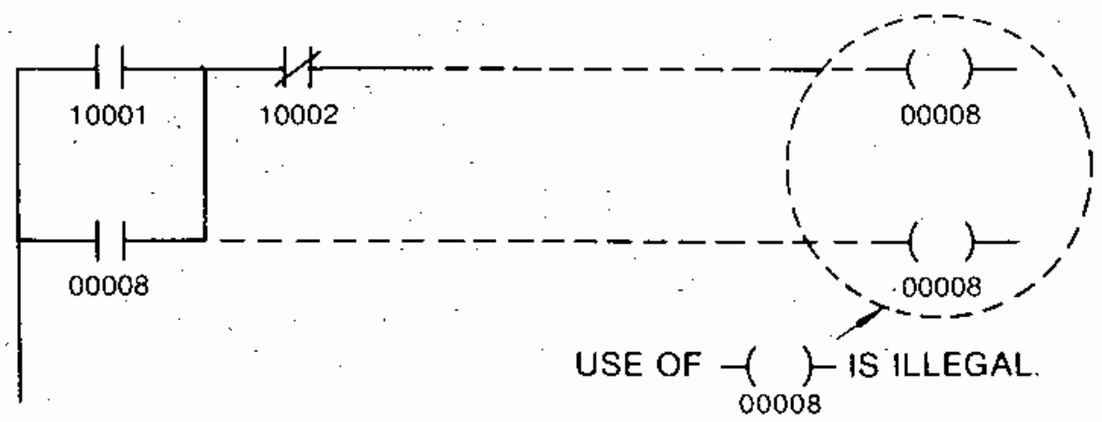


(5) A vetical open is only used to cancel a vertical shunt.

(6) Vertical shunts and opens are not placed at intersections of the 7 lines-by-10 columns matrix and therefore they occupy no memory locations.

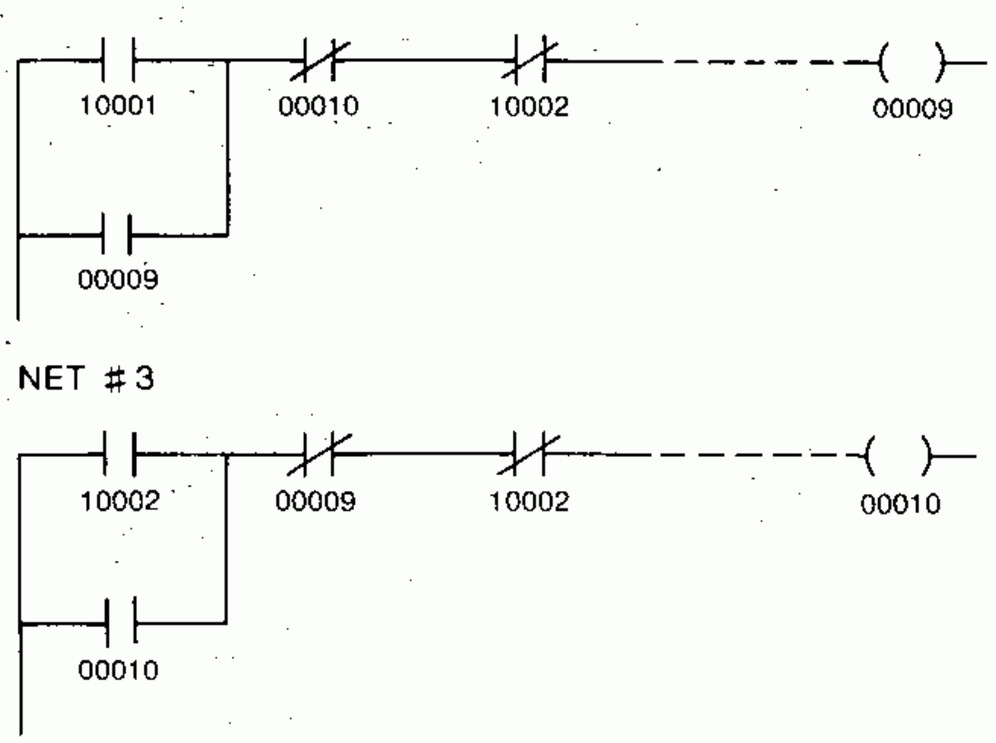
(7) A reference number cannot be given to two or more coils.

Example:



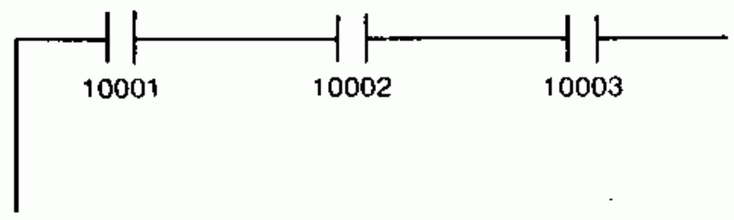
(8) The contact of a coil and input relay (contact) may be used as a NO contact, a NC contact or transitional contact many times.

Example:



(9) A relay circuit without coil is not useful, but not erroneous.

Example:

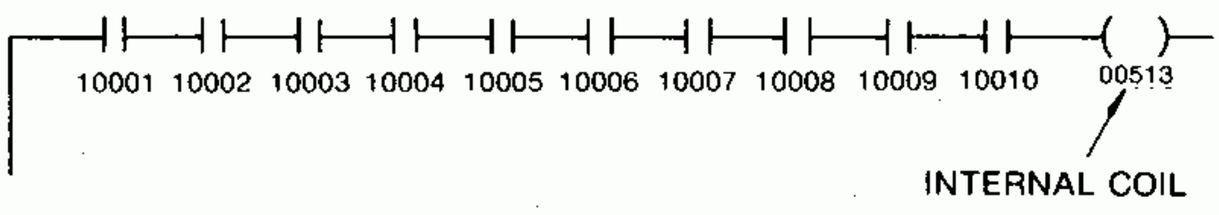


(10) An internal coil is used to extend the number of elements, if more than 10 contacts arranged series or more than seven contacts arranged in parallel are required.

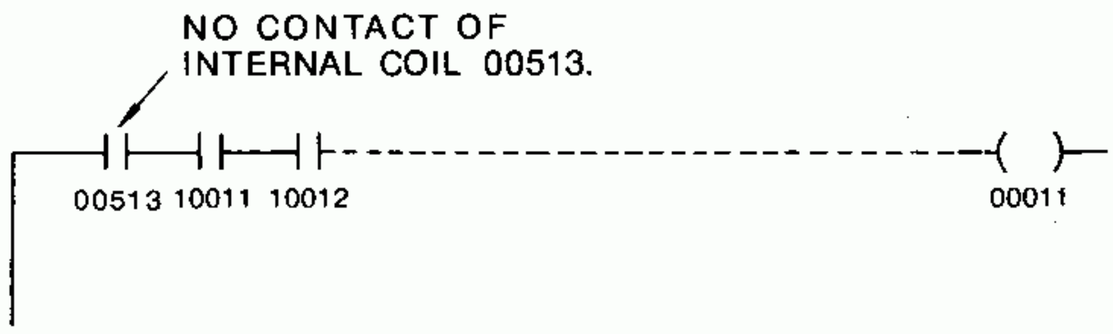
① Examples of series circuit

Examples 1 and 2 show how to make up coil 00011 with a series circuit of NO contacts 10001-10012.

Example 1: Where programmed in two networks



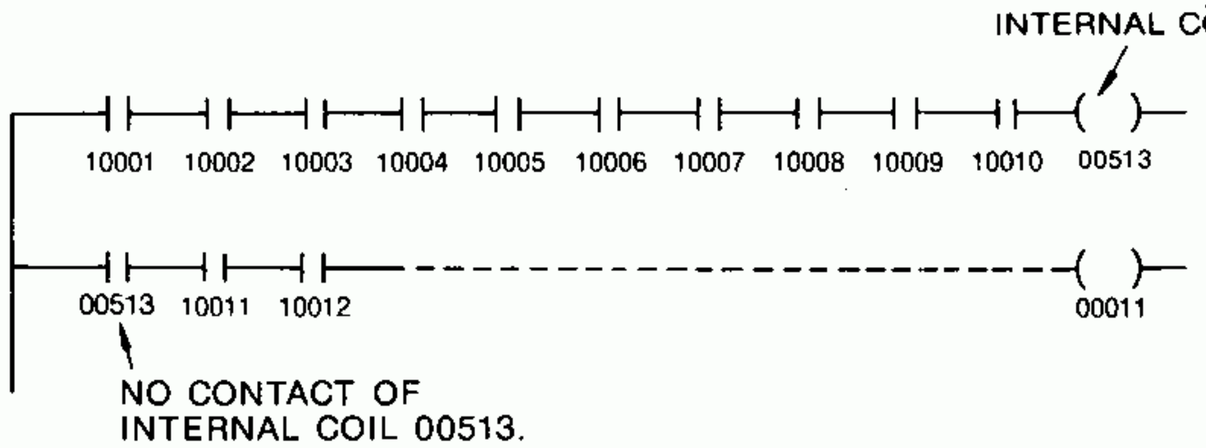
NET #5



NET #6

Coil 00011 is turned on during the scanning cycle when all input relays 10001-10012 all are turned on.

Example 2: Where programmed in one network



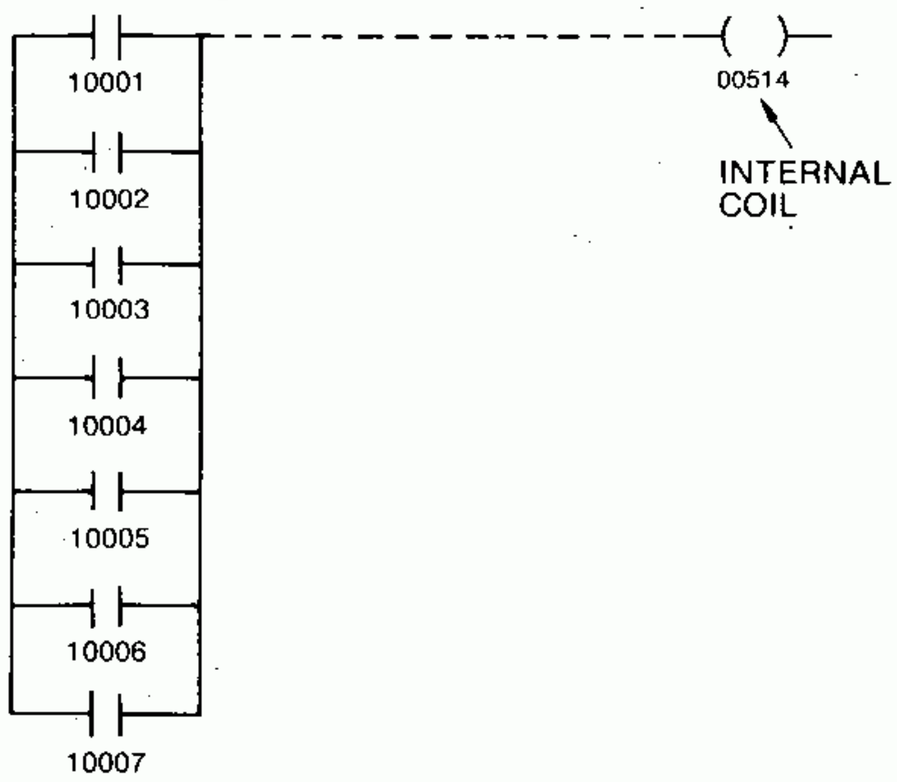
Coil 00513 is turned on during the scanning cycle when all input relays 10001-10010 are turned on, but coil 00011 is turned on during the next scanning cycle (with a delay of one scanning cycle).

5.2.3 Creating of Relay Circuits (Cont'd)

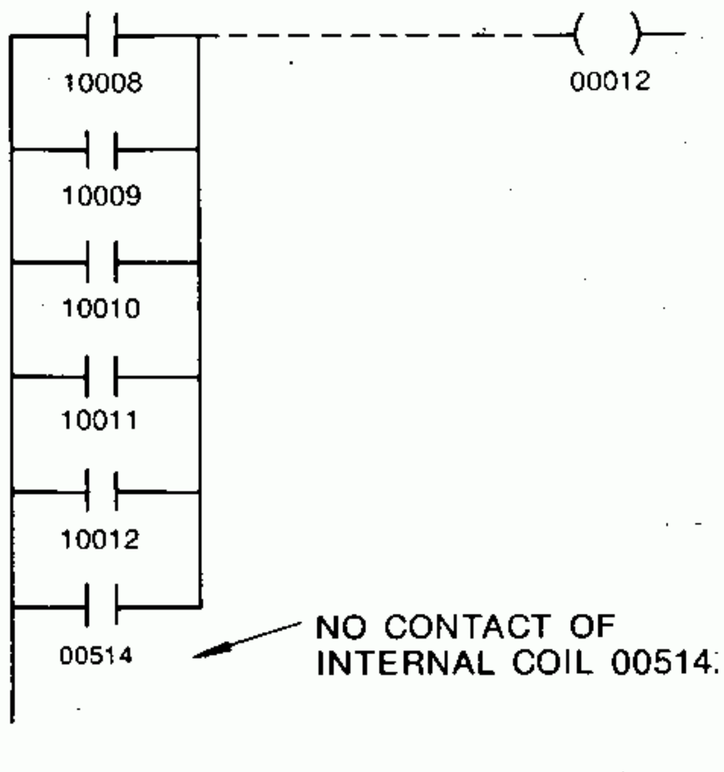
② Examples of Parallel Circuit

Examples 1 and 2 show how to make up coil 00012 with a parallel circuit of NO contacts 10001-10012.

Example 1:

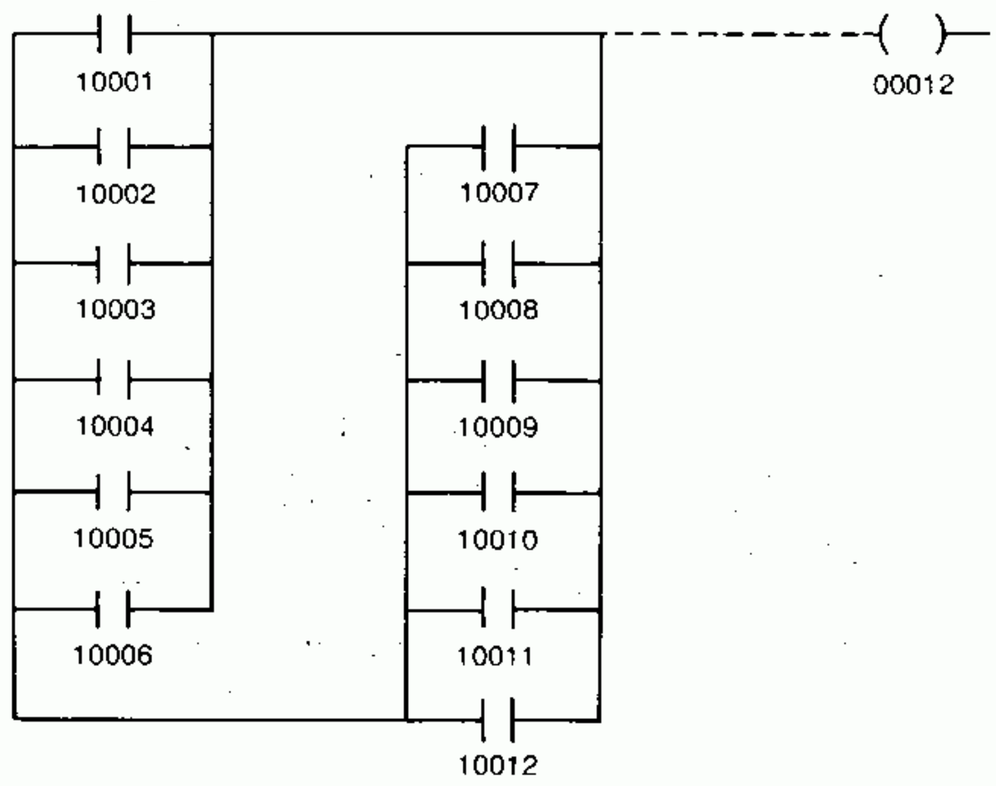


NET #7



NET #8

Example 2:



Note No internal coil is used in this case.

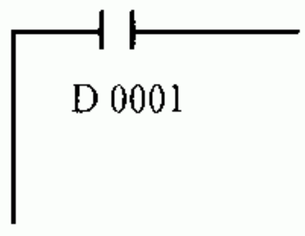
(11) Link Relay

The GL40S enables high speed data link by establishing links with other GL40S's. The states of link coils in other GL40S's can be referenced by using link relays.

Example:

GL40S #1

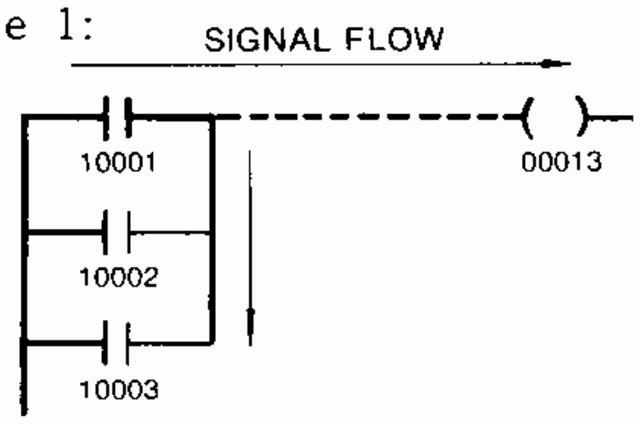
GL40S #2



Synchronized to the link relay D0001 of GL40S #1.

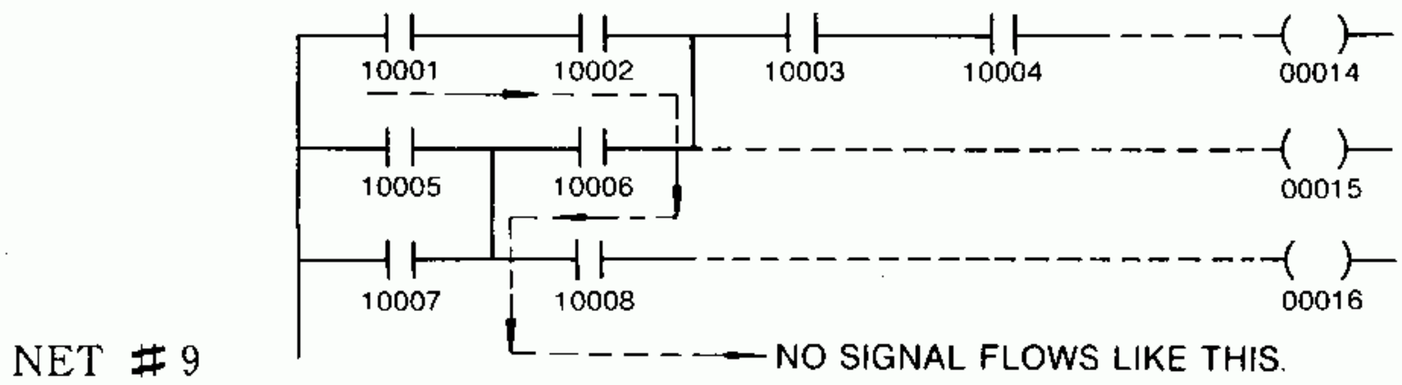
(12) In a relay circuit, a signal always flows from the power rail (left side) to the coils (right side). Vertically it flows either from top to bottom or bottom to top.

Example 1:

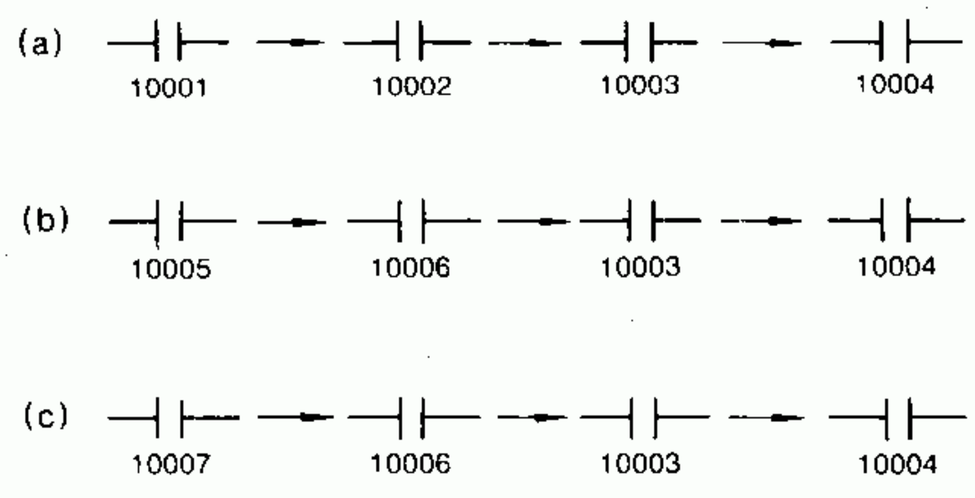


When input relay 10001 is on, a signal flows as shown by arrows and coil 00013 is turned on. On the screen of the P150 programming panel, flows of signal (called "power flows") are indicated by bold lines as shown here.

Example 2:

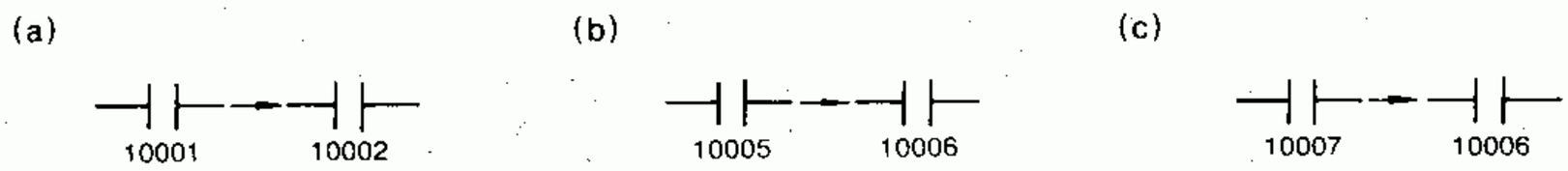


① Paths to turn on coil 00014

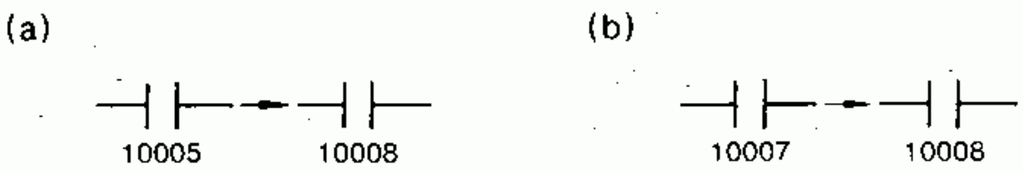


5.2.3 Creating of Relay Circuits (Cont'd)

② Paths to turn on coil 00015



③ Paths to turn on coil 00016

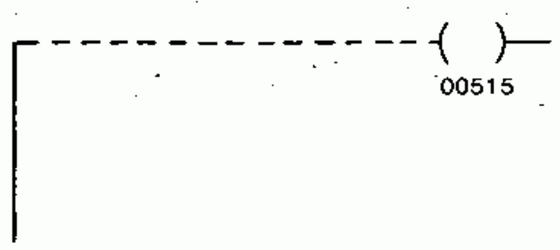


④ Coil 00016 does not turn on through a path as shown with dotted line and therefore it is not necessary to consider a roundabout path.

5.2.4 Sample Application Circuits of Relays

(1) Normally-ON Circuit

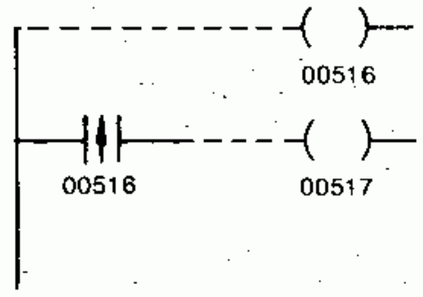
Example:



- Coil 00515 is ON as long as the GL40S is operating properly.
- This is used, for example, in a pulse generating circuit for initialization.

(2) Pulse Generating Circuit for Initialization

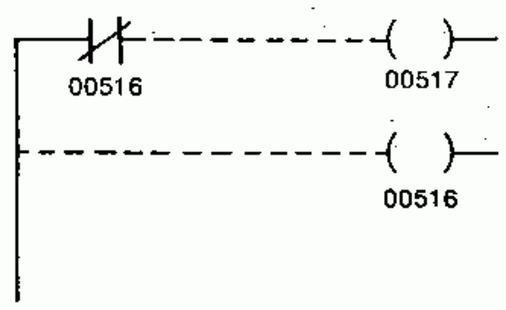
Example 1:



- Coil 00517 is turned on only during the first scanning cycle after the GL40S power is turned on. Examples 1 and 2 are equvalent.
- These are used to set/reset memory circuits, clear the current values of timers and counters, and preset arithmetic constants for initializing the internal logic after the GL40S power is turned on.

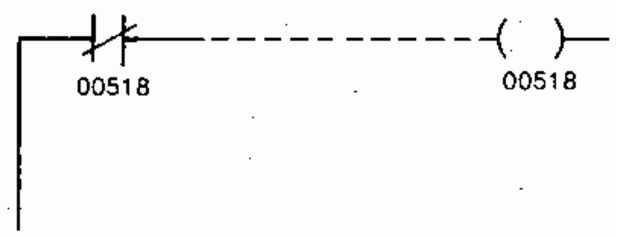
Note Program the circuits of Examples 1 and 2 in a network preceding one using these circuits.

Example 2:



(3) Oscillator

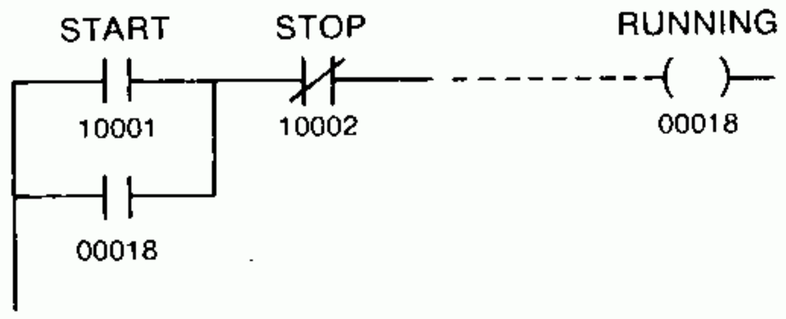
Example:



- Coil 00518 repeats turning on and off every two scanning cycles.
- This is used to perform an arithmetic operation every two scanning cycles.

(4) Self-Holding Circuit (Memory Circuit)

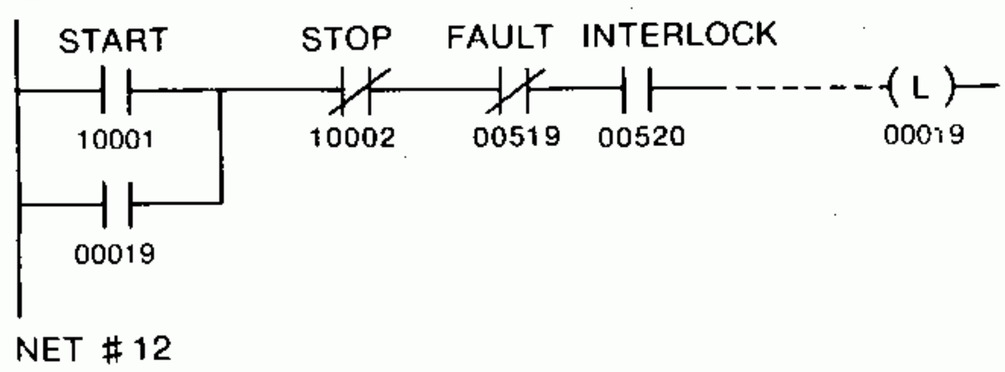
Example:



- As in an ordinary relay circuit, an NO contact of coil is placed in parallel with memory set signal.
- When memory set signal (input relay 10001) comes ON, coil 00018 is turned on and self-held. When memory reset signal (input relay 10002) comes ON, the coil is released from self-holding.
- This is used for start and stop operation with a pushbutton switch which resets automatically.

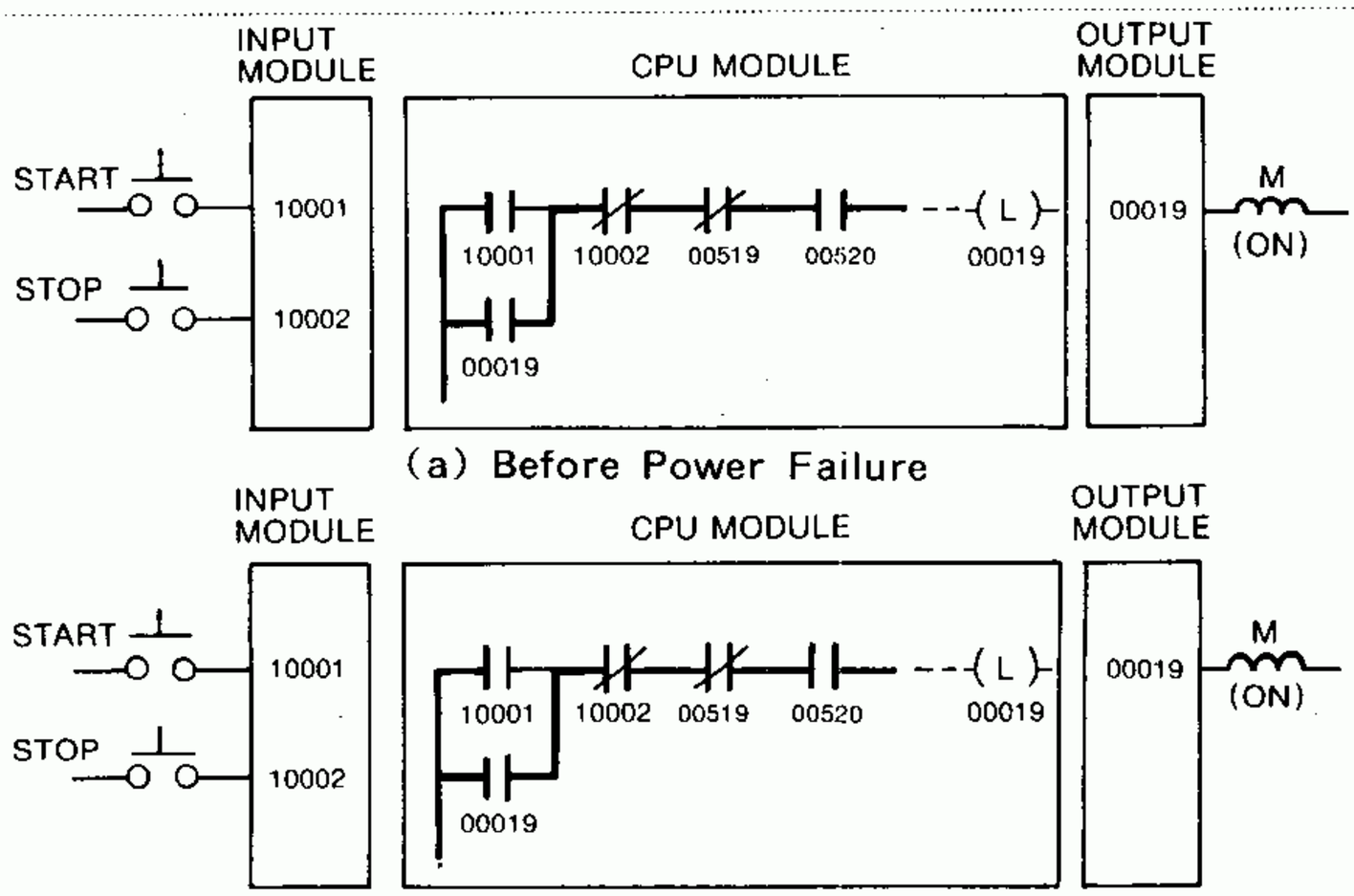
(5) Latched Relay Circuit

(Self-Holding Circuit with Memory during Power Failure)



When a latched coil is used in a self-holding circuit, the status before power failure will be restored after recovery. Any coil can be used as a latched coil.

Example:



Note To use input signal as the reset (OFF) signal (10002 shown above) of a latched relay circuit, read it in as an NO contact then program it as an NC contact (to prevent a latched coil from resetting even if source power of I/O modules stop prior to that of the CPU at power failure).

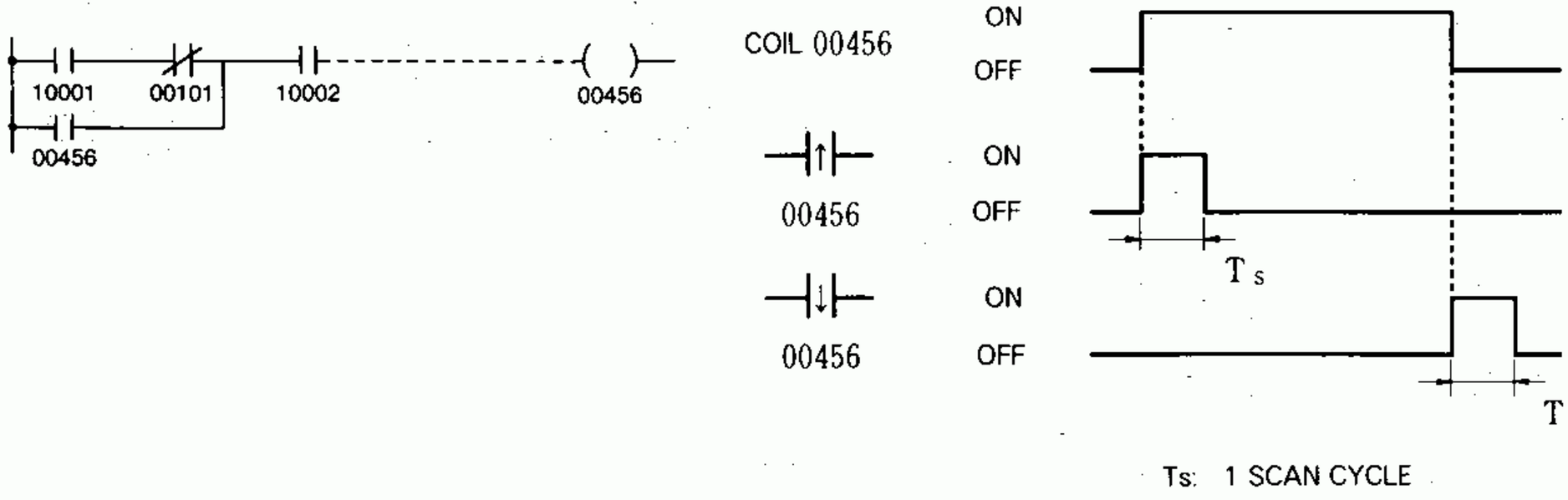
To use a coil as the OFF condition (00519 above) or holding condition (00520 above) of a latched relay circuit, check if it is ON or OFF in reference to the network number during the first scanning cycle after power-on of the GL40S.

For example, if the network of latched coil 00019 appears later (has a greater network number) than that of coil 00519 and earlier than that of coil 00520 in the above example, the latched circuit will be reset if coil 00519 is ON during the first scanning cycle after power-on. It will also be reset unless coil 00520 is a latched coil and ON.

(6) Transitional Contact Circuit

The transitional contact remains on only for one scanning cycle when the associated reference coil has turned on or off. Any input relay or coil may be used as a reference coil.

Note **The transitional contact to the associated reference coil in the skipped network is not operated correctly. (Par. 5.13) The contact remains ON or OFF.**



5.3 TIMERS

5.3.1 Types of Timers

(1) Types

Three types of timers are available as shown in Table 5.6. As many timers as desired used in a range of the program memory capacity and the number of holding registers.

Table 5.6 Types of Timers

Type	Symbol	Unit of Time	Limit of Count
Timer (Seconds)	T1.0	1 sec	1-9,999 sec
Timer (Tenths of Seconds)	T0.1	0.1 sec	0.1-999.9 sec
Timer (Hundredths of Seconds)	T.01	0.01 sec	0.01-99.99 sec

(2) Unit of Time

A timer counts up in certain units of time. The 1-second timer, for example, counts in units of seconds.

(3) Limit of Count

A timer can count up to a certain limit of time. The 1-second timer, for example, counts in the range of 1 to 9,999 seconds. The upper limits of count are presettable.

5.3.2 Timer Configuration

(1) Form

Fig. 5.13 shows the basic form of a timer. A timer is built vertically and needs two elements (top and bottom). Specify any of constant K or reference numbers, referring to Table 5.7. T × × × identifies a specific type of timer. Specify any of T1.0, T0.1, or T.01 referring to Table 5.6.

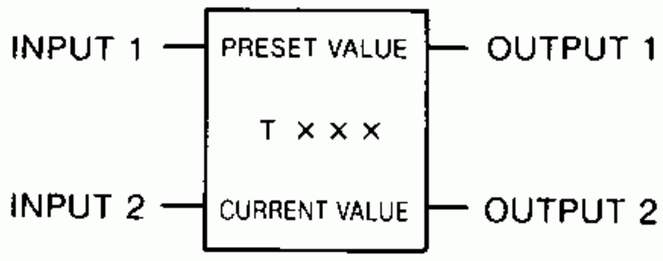


Fig. 5.13 Timer General Form

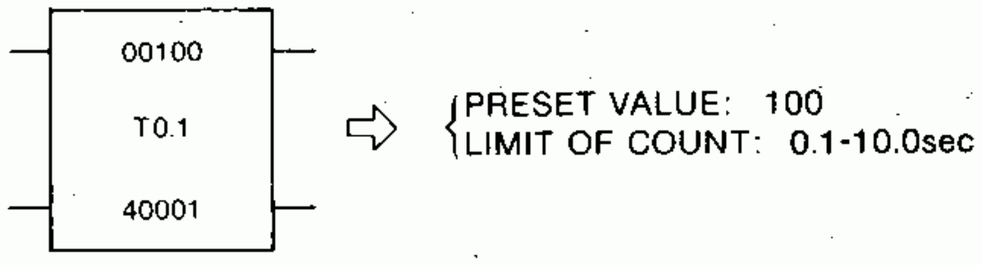
Table 5.7 Timer Elements

Element	Specified Numbers	Description
Top	Constant K (00000-09999) Any of the following: Reference No. Input register (30001-30128) Holding register (40001-42048) Link register (R0001-R1024)	The preset value of timer is constant K, or contents of register reference No.
Bottom	Holding register (40001-42048) Link register (R0001-R1024)	The current value of timer is stored in the specified reference No.

(2) Preset Value

Designate a value of 1 to 9999 to determine the range (variable) of timer.

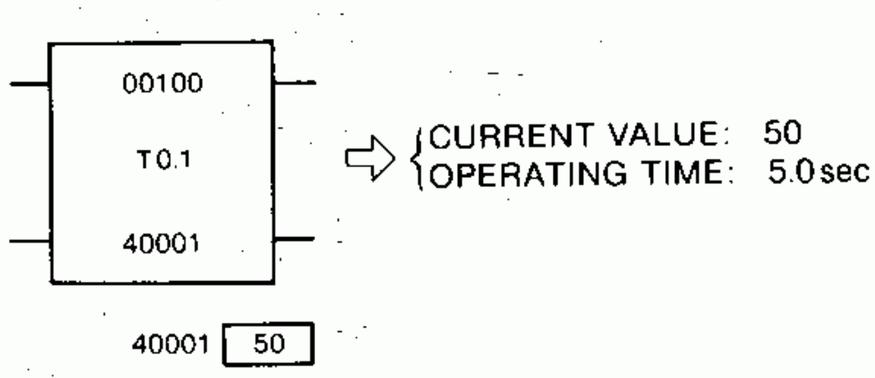
Example:



(3) Current Value

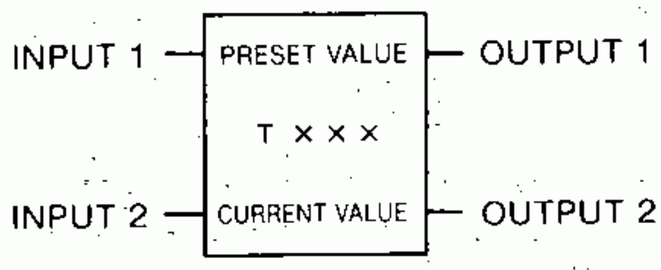
This is the value a timer has counted up.

Example:



5.3.3 Function and Operation of Timer

(1) Timer Function



When input 2 is ON, the timer adds up the time intervals while input 1 is ON. When the current value becomes equal to the preset value, it stops counting with output 1 turned ON and output 2 OFF. When input 2 is OFF, the timer does not count up regardless of the status of input 1. At this time, output 1 is kept OFF and output 2 is ON (the current value is 0).

(2) Timer Operation

Fig. 5.14 and Table 5.8 show the timing chart and operation of the timer.

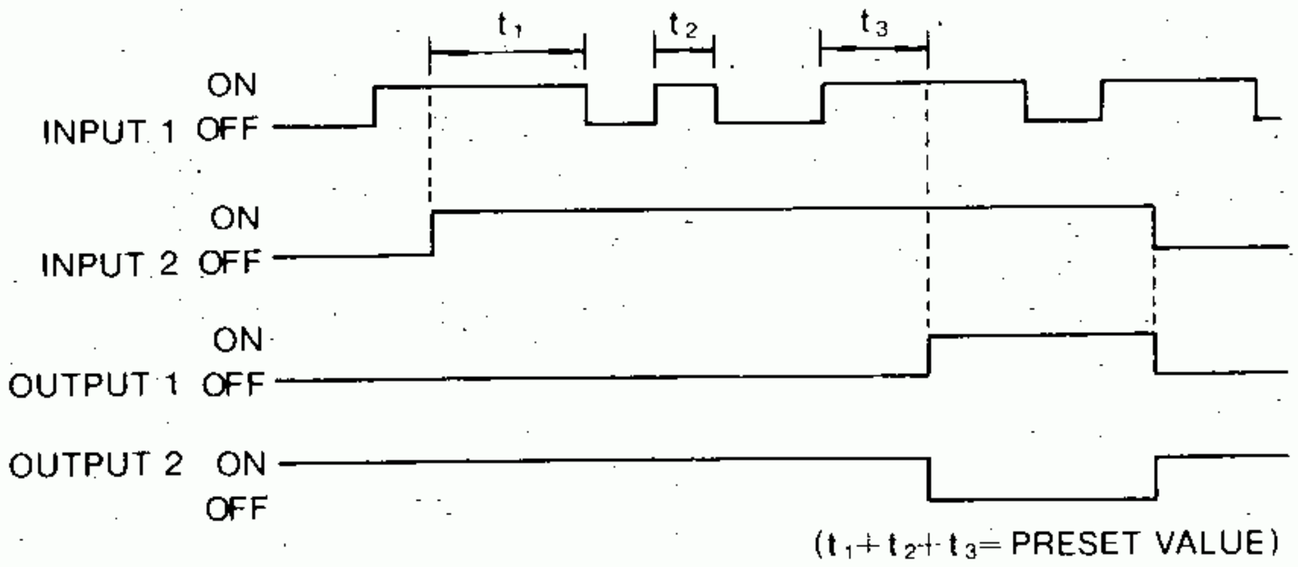


Fig. 5.14 Timer Operation

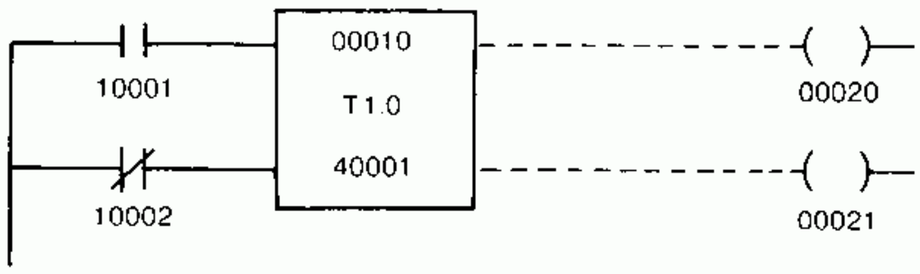
Table 5.8 Timer Operation

Input Status		Timer Status		Current Value	Output Status	
Input 1	Input 2				Output 1	Output 2
–	OFF	Reset status		0	OFF	ON
ON	ON	Operating status	Current value < preset value	Increase	OFF*	ON*
			Current value = Preset value	Preset value	ON	OFF
OFF	ON	Standstill status	Current value < Preset value	Constant	OFF	ON
			Current value = Preset value	Preset value	ON	OFF

*As a result of increasing current value, current value < preset value.

(3) Sample Timer

Example:



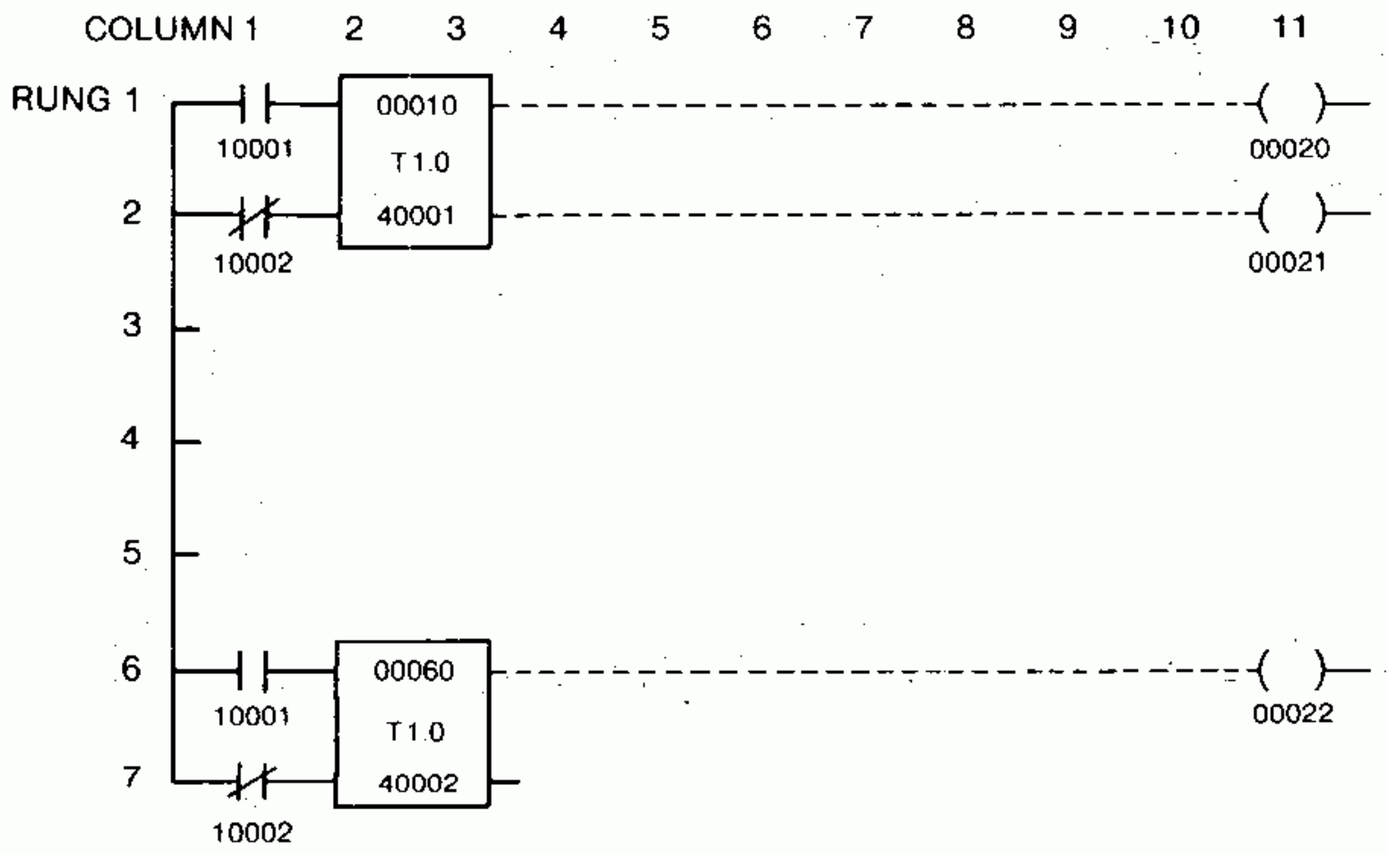
- This timer counts up only when input 2 is ON. (input relay 10002 is OFF).
- If input 1 (input relay 10001) is turned on while input 2 is ON, the timer is operated and the current value (contents of 40001) is increased by one every second.
- When the current value is equal to the preset value (10), the timer stops and output 1 (coil 00020) is turned on and output 2 (coil 00021) off.
- If input 2 is ON and input 1 is cycled ON-OFF-ON-OFF, the time intervals while input 1 is ON are added. See Fig. 5.14.
- When input 2 is OFF, the timer does not count up regardless of the status of input 1. At this time, the current value is 0, output 1 OFF, and output 2 ON.

5.3.4 Programming Timer Circuit and Precautions

(1) Programming Timer Circuit

A timer needs two elements placed vertically (top and bottom) in a network. It can be used at any intersection of the 7 lines-by-10 columns matrix, but the top element (preset value) cannot be located on line 7.

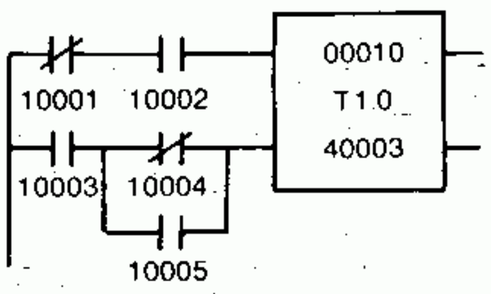
Example:



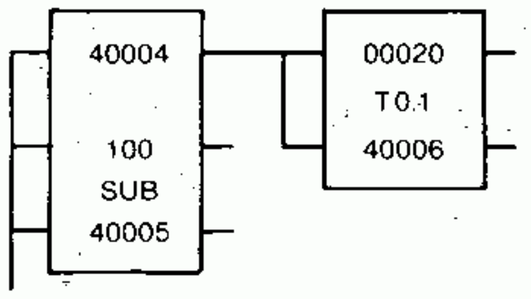
(2) Timer Inputs

Inputs to the timer may be outputs of relays, other timers, counters, arithmetic operations, or data processing circuits.

Example 1:



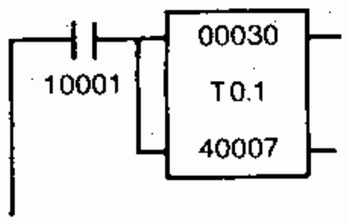
Example 2:



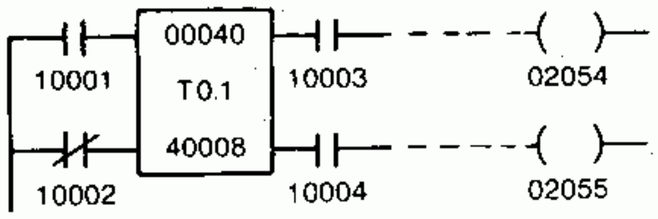
(3) Timer Outputs

Coils may not be connected to outputs 1 and 2 of a timer. It is permitted to insert a relay contact to the right of an output or to connect an output directly to an input of a logic, except relays.

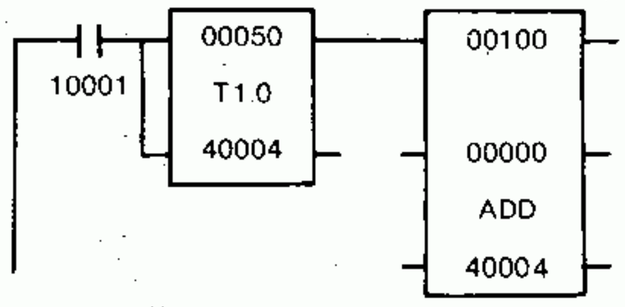
Example 1:



Example 2:



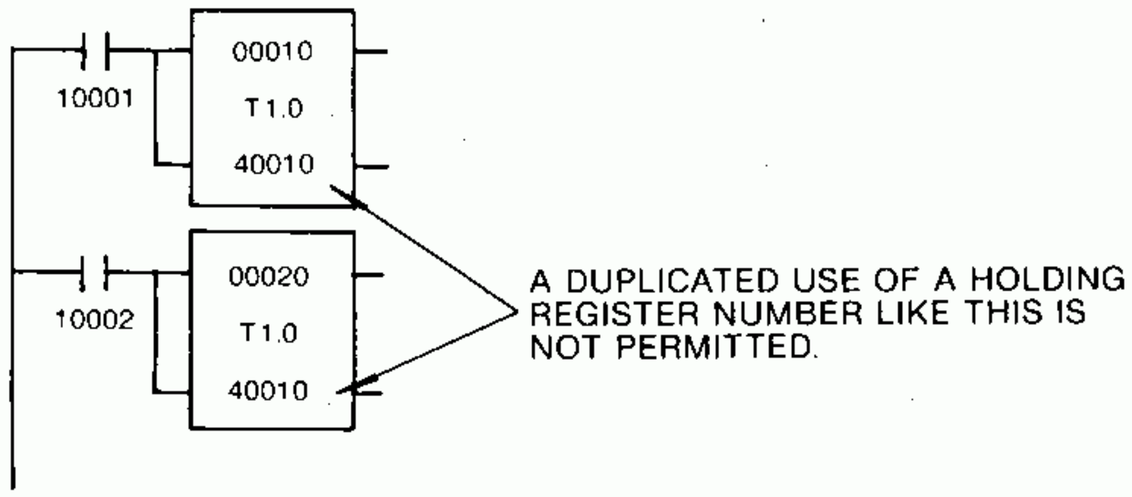
Example 3:



(4) Storing Timer Current Value

The register number of the holding register storing the current value of a timer cannot be the same as the register number of the holding register storing the current value of another timer or counter.

Example:

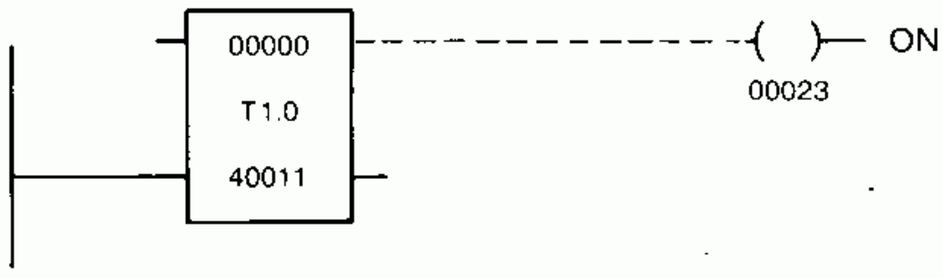


Due to current value changes, the holding register storing the current value of a timer cannot be used as the register storing arithmetic operation results. However, the register can be used as an operand.

(5) Preset Value of Timer

When the preset value of a timer is 0 and input 2 is ON, output 1 is ON (output 2 is OFF), regardless of ON/OFF operation of input 1.

Example:



(6) Relationship of Preset and Current Values

Ordinarily, the current value does not exceed the preset value, but it is possible to make the current value greater than the preset value by arithmetic operation or data transfer function. In such a case, the current value becomes equal to the preset value(output 1 is turned ON) as soon as the timer circuit is solved.

(7) Timer Error

Error of the timer is given as follows.

Maximum value of error = Unit of preset timer + 1 scan time

For example, if T1.0 is used for a 1-second timer, error may rise to 1 second (the timer may reach the limit a second earlier than the real time limit). Therefore, the use of T0.1 or T.01 is recommended in this case.

(8) Current Value of Timer at Power ON

During power failure, the timers of the GL40S memorize the values before power failure. When the GL40S is turned on, the current value of the coil will be reset to 0 or remain unchanged depending on the type and status of the reference coil of contact used for input 2. Table 5.9 shows these relationships.

Table 5.9 Current Value of Timer at Power ON

Signal to Input 2				Current Value at Power ON
Type of Reference Coil		Status	Contact	
Input Relay Latched Coil		ON	NO	Value at power failure
			NC	0
		OFF	NO	0
			NC	Value at power failure
Coil (Not including Latched Coil)	n ≦ m	ON	NO	Value at power failure
			NC	0
		OFF	NO	0
			NC	Value at power failure
	n > m	OFF	NO	0
			NC	Value at power failure

Note
1. The number of the network including the reference coil of signal to input 2 is n and the number of the network including the timer circuit is m.
2. If the signal to input 2 is composed of more than one contact, obtain the input 2 status from each contact status.
3. The current value of the timer, at the GL40S power ON, is the value read when the timer circuit has been solved during the first scanning cycle after power is applied to the GL40S.

5.3.5 Application Timer Circuits

On-delay and off-delay timers can be obtained by vertically shunting timer inputs 1 and 2. Various timer circuits can be realized by using different signal for inputs 1 and 2.

(1) ON-delay Timer

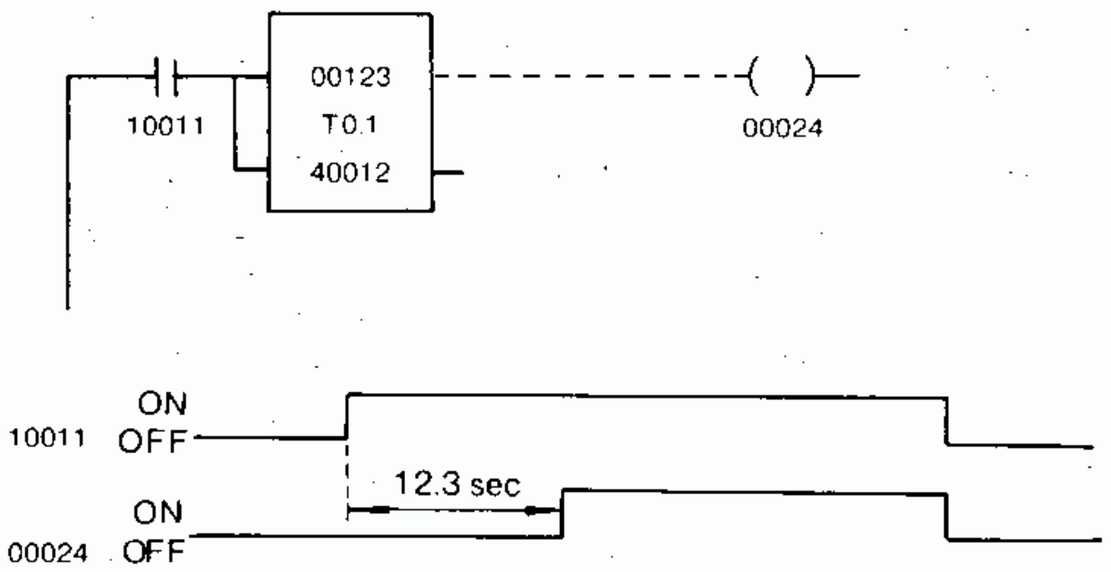
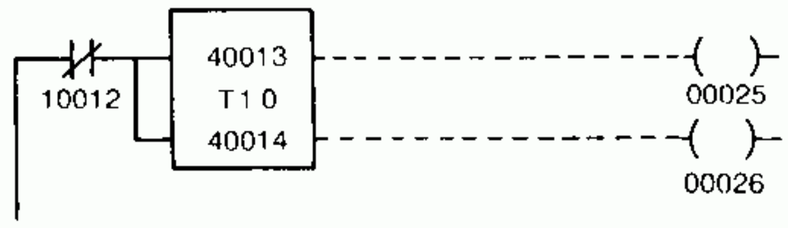


Fig. 5.15 Sample ON-delay Timer

(2) OFF-delay Timer



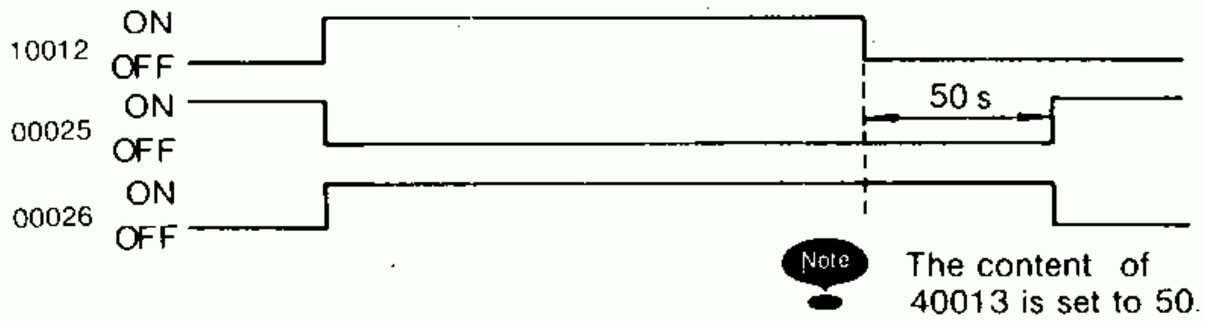
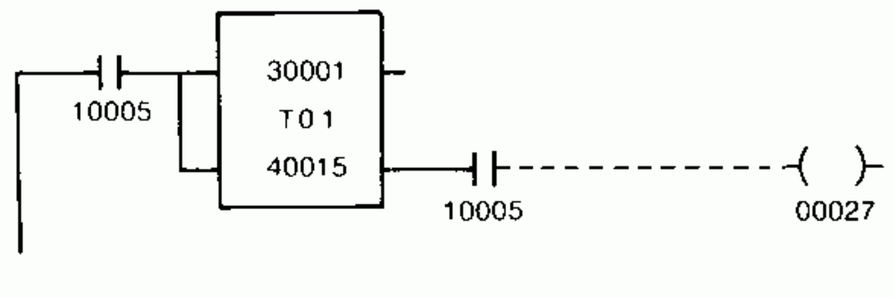


Fig. 5.16 Sample
OFF-delay Timer

(3) One-shot Timer



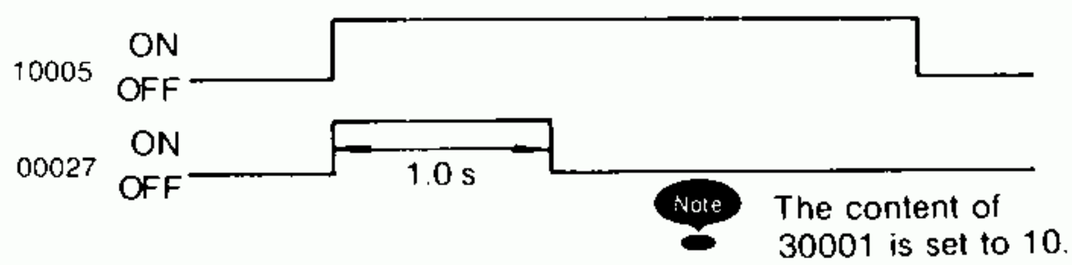
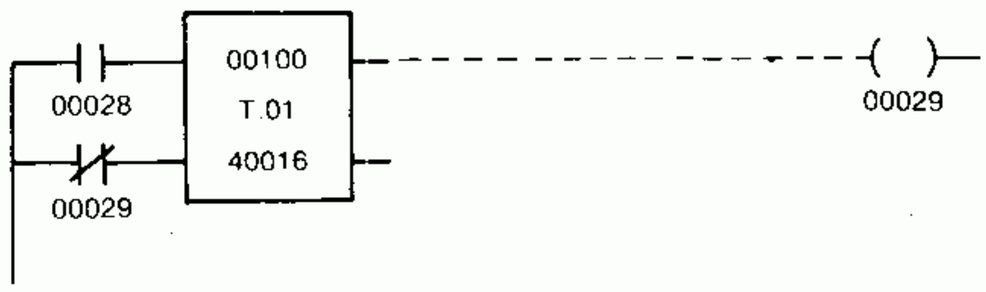


Fig. 5.17 Sample
One-shot Timer

(4) Pulse Generating Circuit



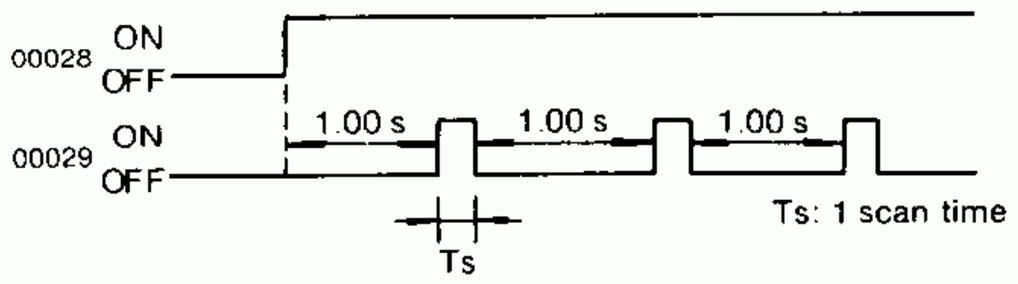
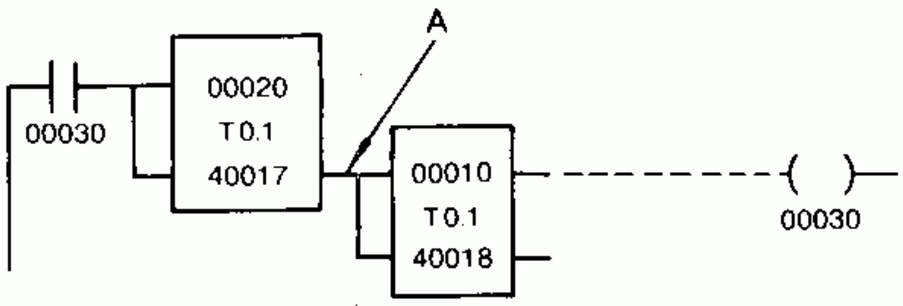


Fig. 5.18 Sample Basic
Pulse Generating Circuit

(5) Flicker Relay



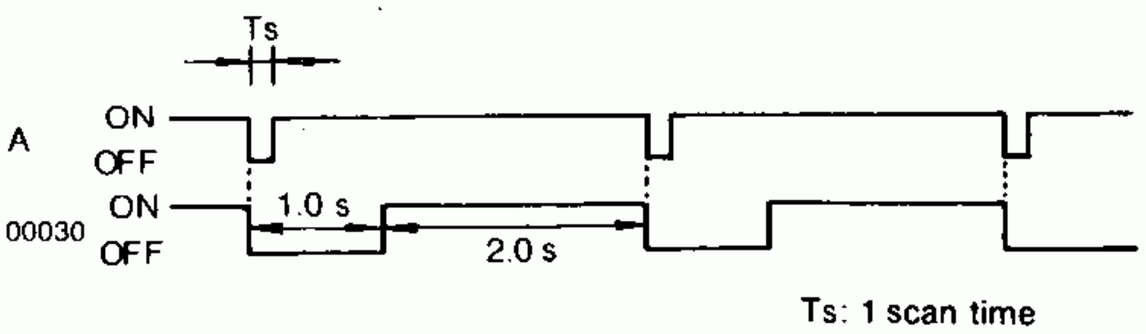


Fig. 5.19 Sample
Flicker Relay

5.4. COUNTERS

5.4.1 Types of Counters

(1) Types

Two types of counters are available as shown in Table 5.10. As many counters as desired may be used, in a range of the program memory capacity and the number of holding registers.

Table 5.10 Types of Counters

Type	Symbol	Unit of Count	Limit of Count
Up Counter	UCTR	1 pulse	1-9,999 pulses
Down Counter	DCTR		

(2) Unit of Count

An up or down counter increases or decreases the current count by one pulse.

(3) Limit of Count

An up or down counter increases or decreases in a range of pulses counted (1 to 9,999 pulses). The upper limits of count are presettable.

5.4.2 Counter Configuration

(1) Form

Fig. 5.20 shows the basic form of a counter. A counter needs two elements placed vertically (top and bottom). Specify any of constant K or reference numbers referring to Table 5.11. XCTR identifies a specific type of counter. Specify UCTR or DCTR in reference to Table 5.10.

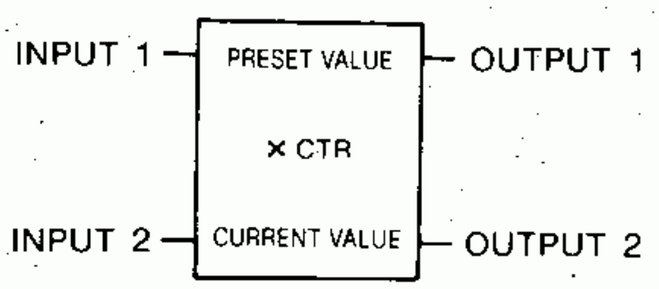


Fig. 5.20 Counter General Form

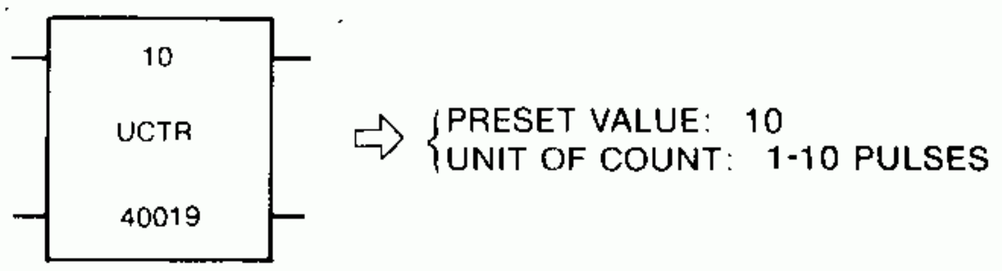
Table 5.11 Counter Elements

Element	Specified Numbers	Description
Top	Constant K (00001-09999) Any of the following: Input register (30001-30128) Holding register (40001-42048) Link register (R0001-R1024)	The preset value of counter is constant K, or contents of register reference No.
Bottom	Holding register (40001-42048) Link register (R0001-R1024)	The current value of counter is stored in the specified reference.

(2) Preset Value

Specify a value of 1 to 9999 to determine the upper limit (variable) of count.

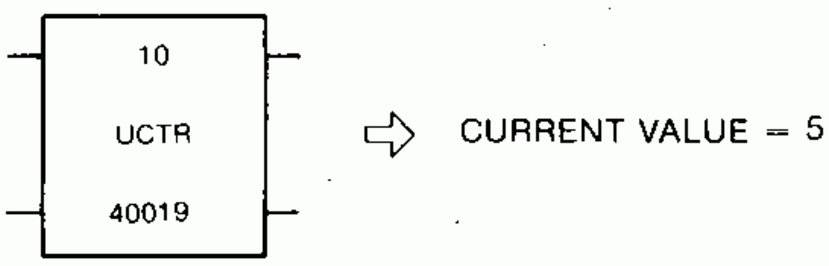
Example:



(3) Current Value

This is the value a counter has counted.

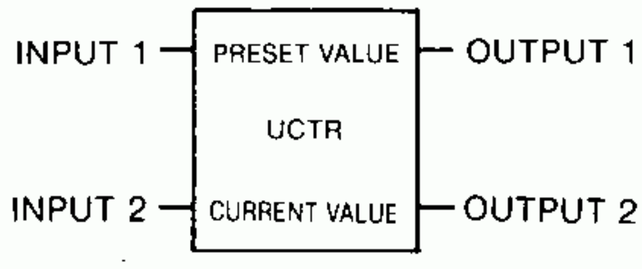
Example:



5.4.3 Function and Operation of Counter

5.4.3.1 Up Counter

(1) Up Counter Function



- When input 2 is ON, the up counter counts the number of times in which input 1 is turned from OFF to ON. The current value is increased by one every counting.
- When the current value becomes equal to the preset value, the up counter stops counting with output 1 turned ON and output 2 OFF.
- When input 2 is OFF, the up counter does not count even if input 1 is changed from OFF to ON. At this time, the current value is 0, output 1 is OFF and output 2 ON.

(2) Up Counter Operation

Fig. 5.21 and Table 5.12 show the timing chart and operation of the up counter.

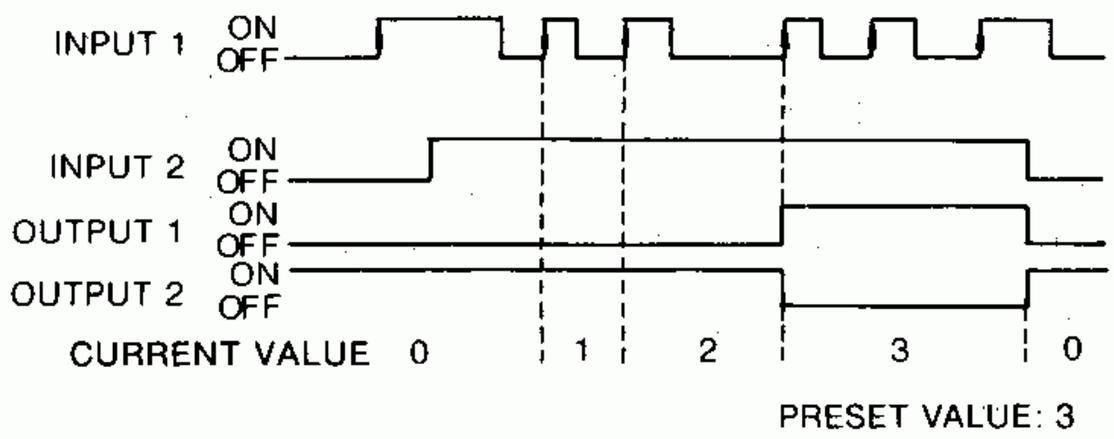


Fig. 5.21 Up Counter Operation

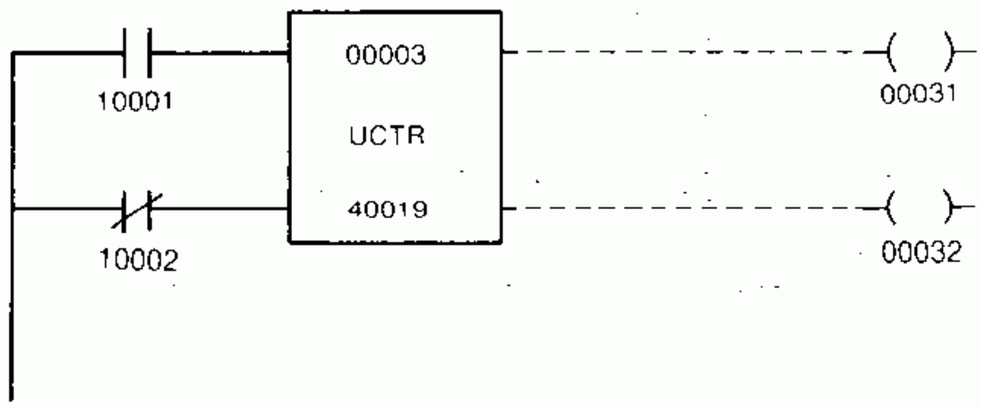
5.4.3 Function and Operation of Counter (Cont'd)

Table 5.12 Up Counter Operation

Input Status		Counter Status		Current Value	Output Status	
Input 1	Input 2				Output 1	Output 2
• ON Status • OFF Status • OFF→ON • ON→OFF	OFF	Reset status		0	OFF	**ON**
OFF→ON	ON	Operating status	Current value < Preset value	Increase (+1)	OFF*	**ON***
			Current value = Preset value	Preset value	**ON**	OFF
ON→OFF	ON	Standstill status	Current value < Preset value	Constant	OFF	**ON**
			Current value = Preset value	Preset value	**ON**	OFF

* As a result of increasing current value, current value < preset value.

(3) Sample Up Counter



(a) The up counter counts only when input 2 is ON (input relay 10002 is OFF).

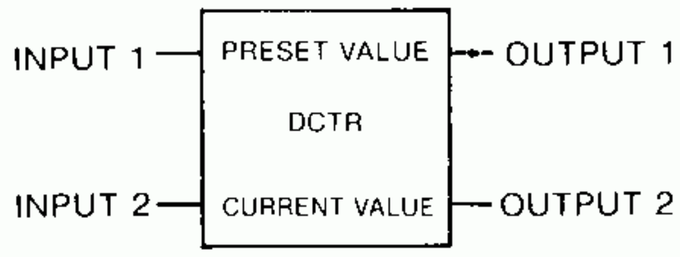
(b) When input 1 (input relay 10001) is turned from OFF to ON while input 2 is ON, the up counter counts and increases the current value (contents of 40019) by one.

(c) When the current value becomes equal to the preset value (3), the up counter stops counting and the output 1 (coil 00031) is turned ON and output 2 (coil 00032) OFF.

(d) When input 2 is OFF (input relay 10002 is ON), the up counter does not count even if input 1 is changed from OFF to ON. At this time, the current value is 0, output 1 OFF, and output 2 ON.

5.4.3.2 Down Counter

(1) Down Counter Function



- When input 2 is ON, the down counter counts the number of times in which input 1 is turned from OFF to ON. The current value is decreased by one every counting.
- When the current value becomes zero, the down counter stops counting with output 1 turned ON and output 2 OFF.
- When input 2 is OFF, the down counter does not count even if input 1 is changed from OFF to ON, At this time, the current value becomes equal to the preset value with output 1 turned OFF and output 2 ON.

(2) Down Counter Operation

Fig. 5.22 and Table 5.13 show the timing chart and operation of the down counter.

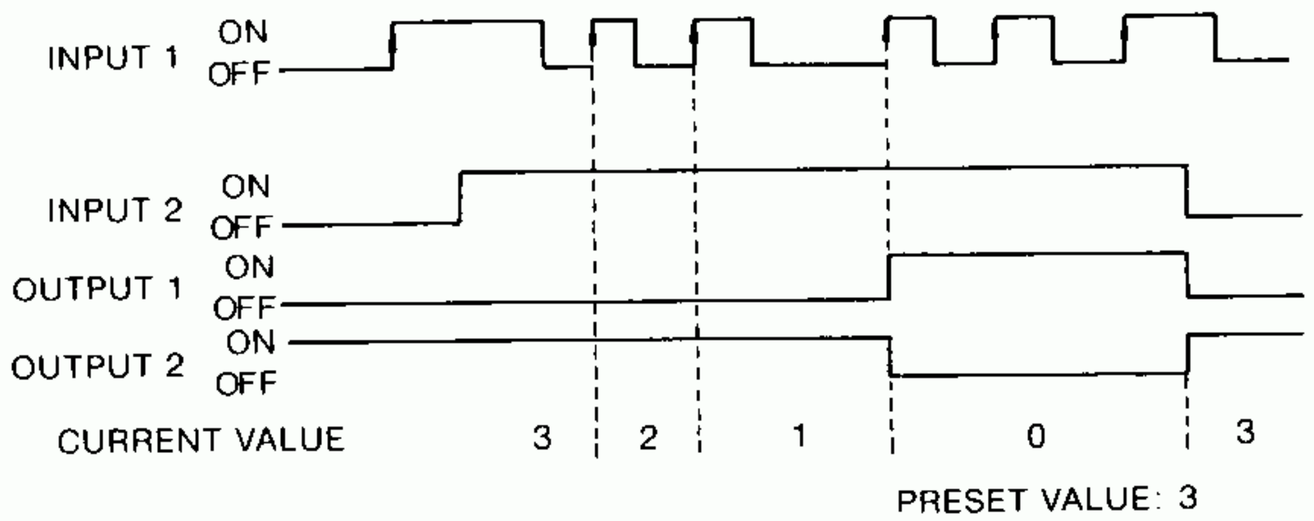


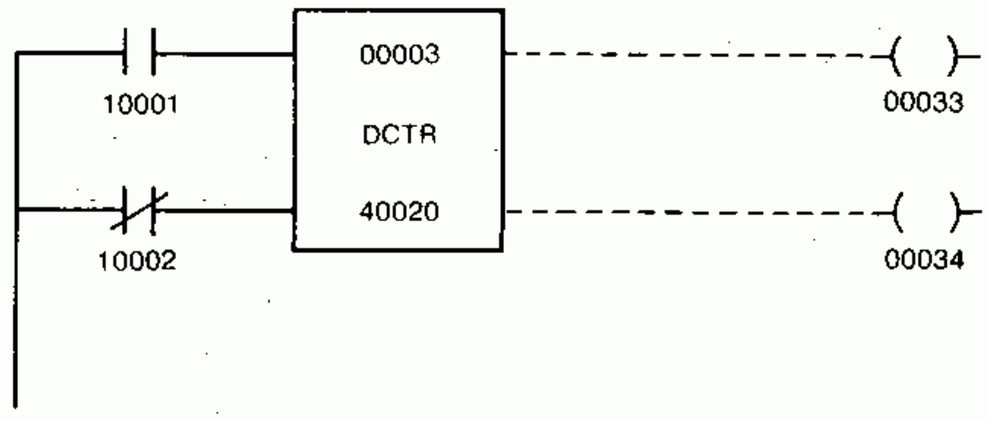
Fig. 5.22 Timing Chart of Down Counter

Table 5.13 Down Counter Operation

Input Status		Counter Status		Current Value	Output Status	
Input 1	Input 2				Output 1	Output 2
• OFF Status • ON Status • OFF→ON • ON→OFF	OFF	Reset status		Preset value	OFF	
OFF→ON	ON	Operating status	Current value > 0	Decrease (-1)	OFF*	
			Currrent vale = 0	0		OFF
• OFF Status • ON Status • ON→OFF	ON	Standstill status	Current value > 0	Constant	OFF	
			Current value = 0	0		OFF

* As a result of decreasing current value, current value > 0.

(3) Sample Down Counter



NET #21

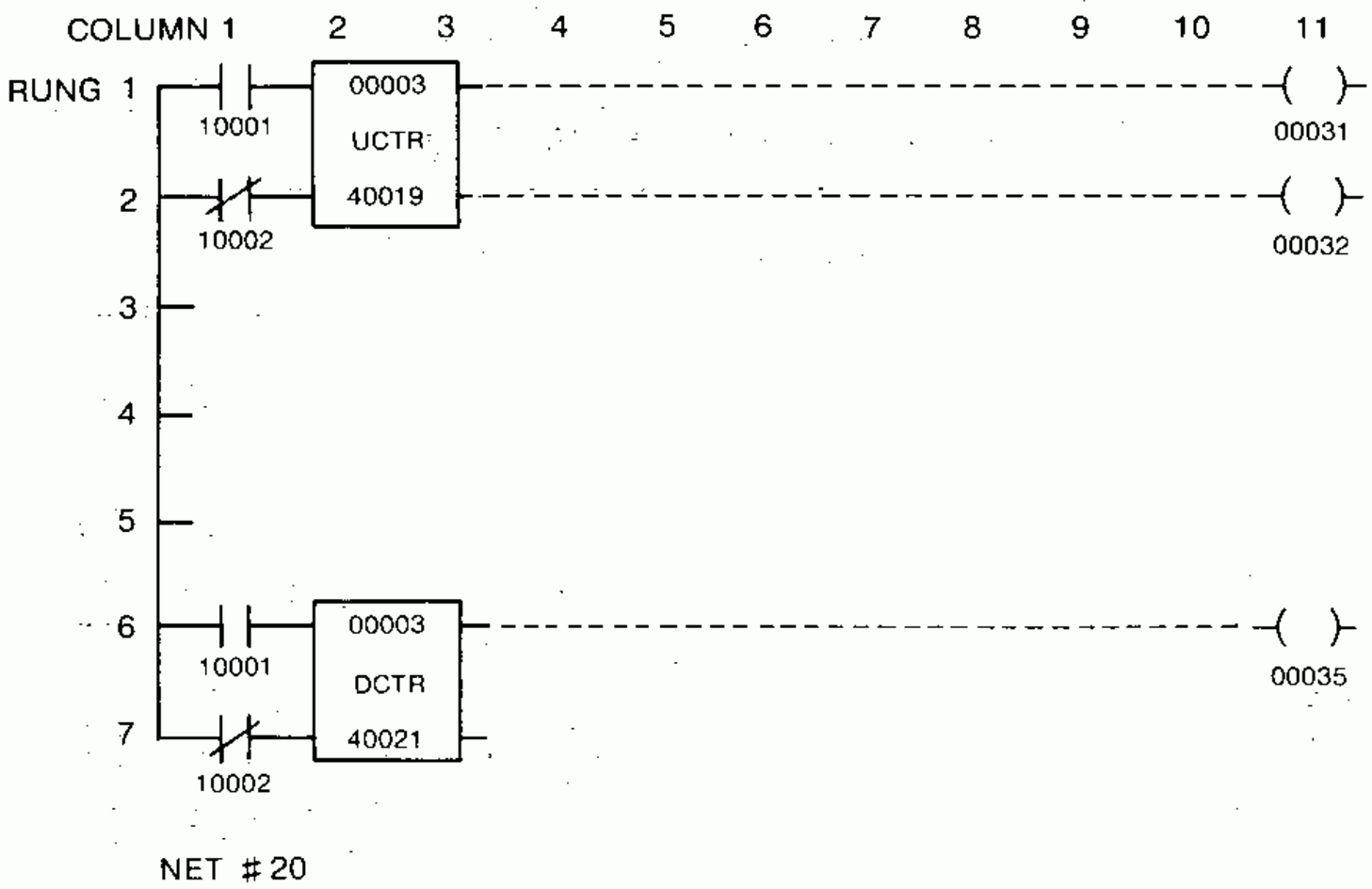
- The down counter counts only when input 2 is ON (input relay 10002 is OFF).
- When input 1 (input relay 10001) is turned from OFF to ON while input 2 is ON, the down counter decreases the current value (contents of 40020) by one.
- When the current value becomes zero, the down counter stops to counting and output 1 (coil 00033) is turned ON and output 2 (coil 00034) OFF.
- When input 2 is OFF (input relay 10002 is ON), the down counter does not count even if input 1 is changed from OFF to ON. At this time, current value becomes equal to the preset value and output 1 is turned OFF and output 2 ON.

5.4.4 Programming Counter Circuit and Precautions

(1) Programming Counter Circuit

An up and down counter occupies two elements placed vertically (top and bottom) in a network. It can be used at any intersection of the 7 lines-by-10 columns matrix, but the top element (preset value) cannot be located on line 7.

Example:

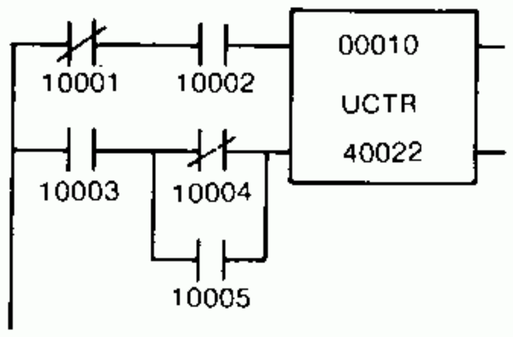


NET #20

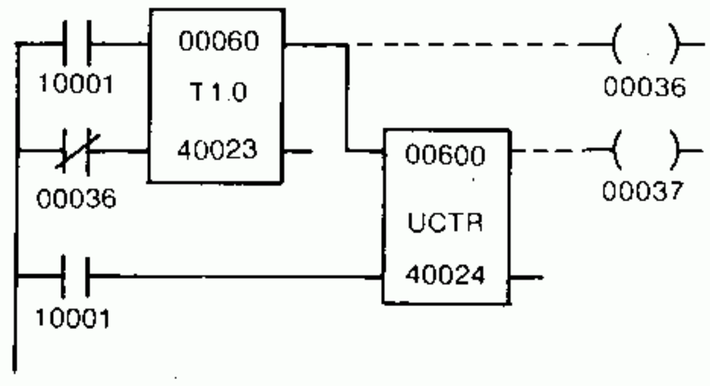
(2) Counter Inputs

Inputs to the counter may be outputs of relays, other counters, timers, arithmetic operations, and data processing circuits.

Example 1:



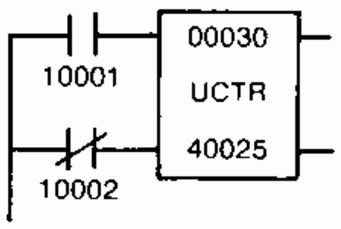
Example 2:



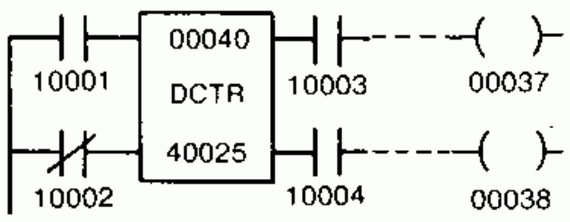
(3) Counter Outputs

Coils may not be connected to outputs 1 and 2 of a counter. It is permitted to insert a realy contact to the right of an output or to connect an output directly to an input of logic, except relays.

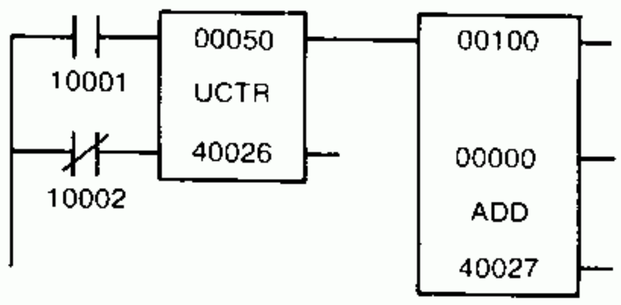
Example 1:



Example 2:



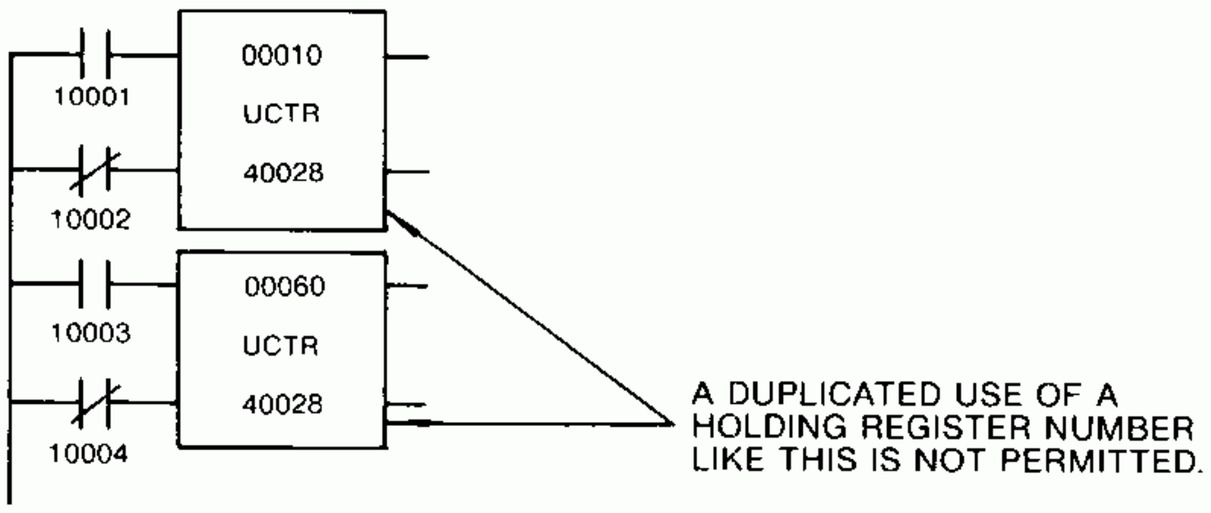
Example 3:



(4) Storing Counter Current Value

The register number of the holding register storing the current value of a counter cannot be the same as the register number of the holding register storing the current value of another counter or timer. In special cases when used as an up/down counter (see (5) of Application Circuits of Counter), a register number may be used for different counters or timers.

Example:

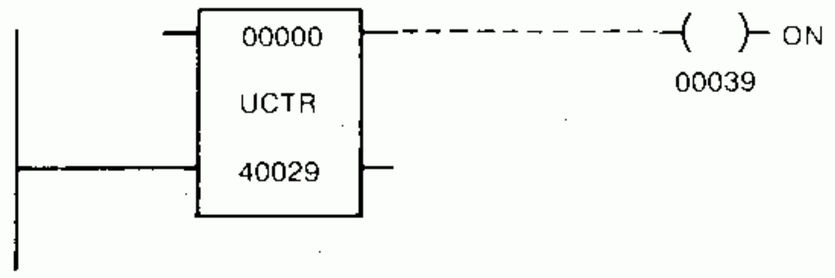


Due to current value changes, the holding register storing the current value of a timer cannot be used as the register storing arithmetic operation results. However, the register can be used as an operand.

(5) Preset Value of Counter

When the preset value of an up- or down-counter is 0 and input 2 is ON, output 1 is ON (output 2 is OFF), regardless of ON/OFF operation of input 1.

Example:



(6) Relationship of Preset and Current Values

Ordinarily, the current value does not exceed the preset value, but it is possible to make the current value greater than the preset value by arithmetic operation or data transfer function. In such a case, the current value becomes equal to the preset value as soon as the counter circuit is solved.

(7) Input Pulse Width

When operating a counter by signals (count pulses) fed from an external device, one scan time or more is required for each pulse width (ON or OFF). In addition, delay of response of the input module which receives the input pulses must be taken into consideration.

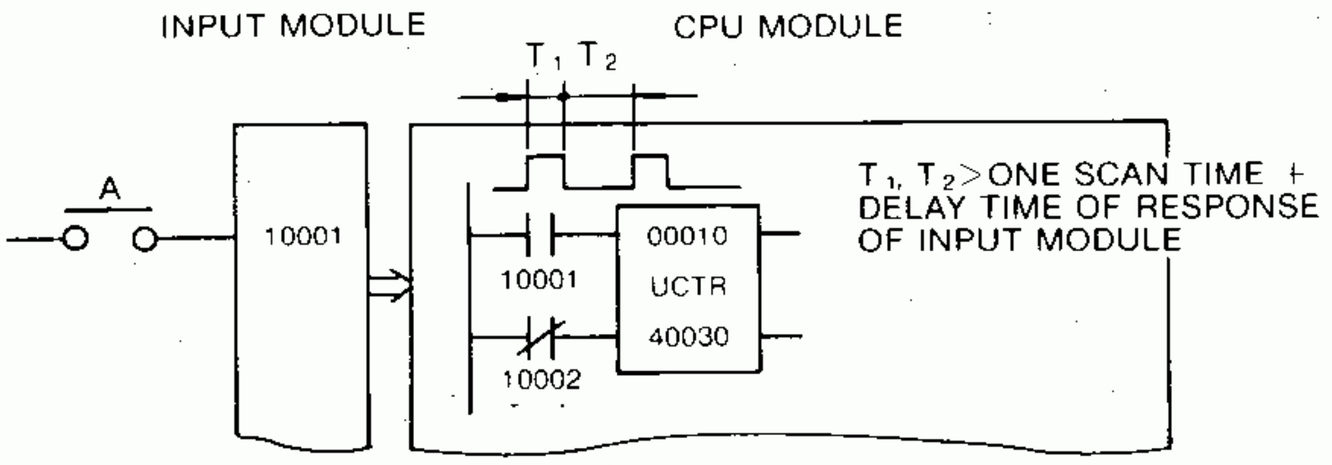


Fig. 5.23 Counter Operation by Singnals Fed from an External Device

(8) Current Value of Counter at Power ON

Like the timer, the counter keeps the current value even if power fails. When the GL40S is turned ON, the current value is determined by signals for input 2.

Table 5.14 Current Value of Counter at Power ON

Signal to Input 2				Current Value at Power ON	
Type of Reference Coil		Status	Contact	Up Counter	Down Counter
Input Relay Latch Coil		ON	NO	Value at power failure	Value at power failure
			NC	0	Preset value
		OFF	NO	0	Preset valur
			NC	Value at power failure	Value at power failure
Coil (Not including Latch Coil)	n ≦ m	ON	NO	Value at power failure	Value at power failure
			NC	0	Preset value
		OFF	NO	0	Preset value
			NC	Value at power failure	Value at power failure
	n > m	OFF	NO	0	Preset value
			NC	Value at power failure	Value at power failure

Note
1. The number of the network inculding the reference coil of signal to input 2 is n and the number of the network including the counter circuit is m.
2. If the signals to input 2 is composed of more than one contact, obtain the input 2 status from each contact status.
3. The current value of the counter at the GL40S power ON, is the value read when the counter circuit has been solved during the first scanning cycle after power is supplied to the GL40S.

5.4.5 Application Counter Circuits

(1) Large-capacity Counter

- A large-capacity counter (having a preset value of 10000 or more) can be created by combining multiple counters.
- Fig. 5.24 shows a large-capacity counter (preset value = 100 × 150 = 15000) using two up counters.

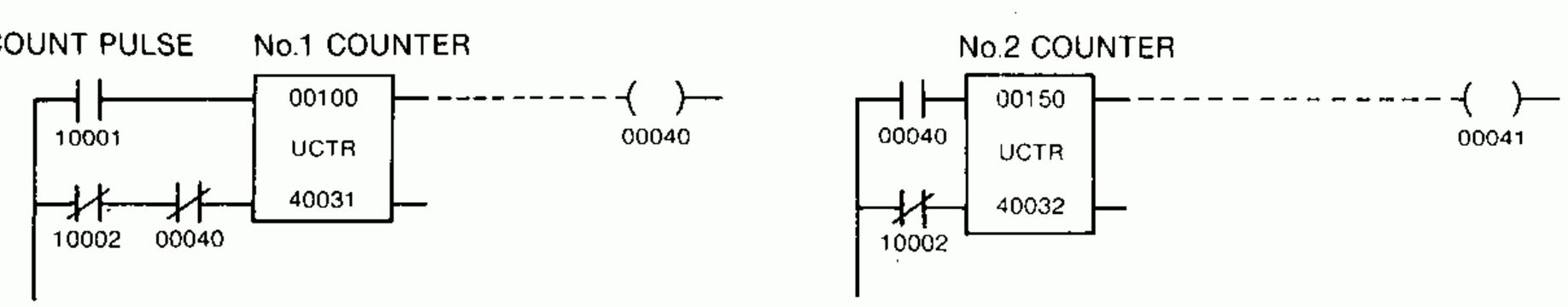


Fig. 5.24 Large-capacity Counter

- The No. 1 counter generates a pulse (ON time = 1 scan time) for the coil 00040 for every 100 input pulses.
- When the No. 2 counter counts 150 pulses output by the No. 1 counter, coil 00041 is turned ON to indicate that 15000 pulses (100 × 150) have been counted.
- If the contents of 40031 is x (0 ≦ x ≦ 100) and the contents of 40032 is y (0 ≦ y ≦ 150), the current value (pulse count) of the large-capacity counter is x + 100y.

(2) Long-time Timer

- A long-timer (having a preset value of 10000 or more) can be created by combining a timer and an up counter.
- Fig. 5.25 shows a long-time timer (preset value = 60 × 600 = 36000) using a 1-second timer and an up counter.

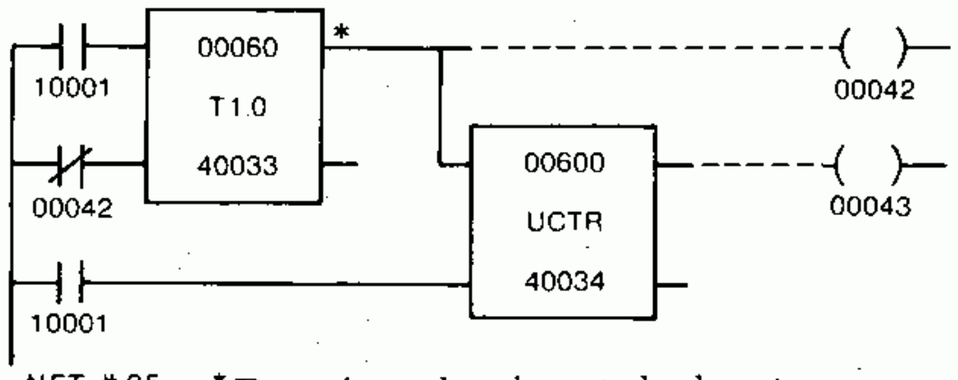


Fig. 5.25 Long-time Timer

- The 1-second timer generates a pulse (ON time = 1 scan time) for coil 00042 for every 60 seconds while input relay 10001 is ON.
- When the up counter counts 600 pulses output by the 1-second timer, coil 00043 is turned ON to indicate that 36000 seconds (60×600) have elapsed.
- If the contents of 40033 is x ($0 \leqq x \leqq 60$) and the contents of 40034 is y ($0 \leqq y \leqq 600$), the current value of the long-timer is $x + 60y$ (the number of seconds the timer has operated).

(3) Clock

- A clock circuit can be created by combining a timer with an up counter.
- Fig. 5.26 shows an example of a clock circuit.

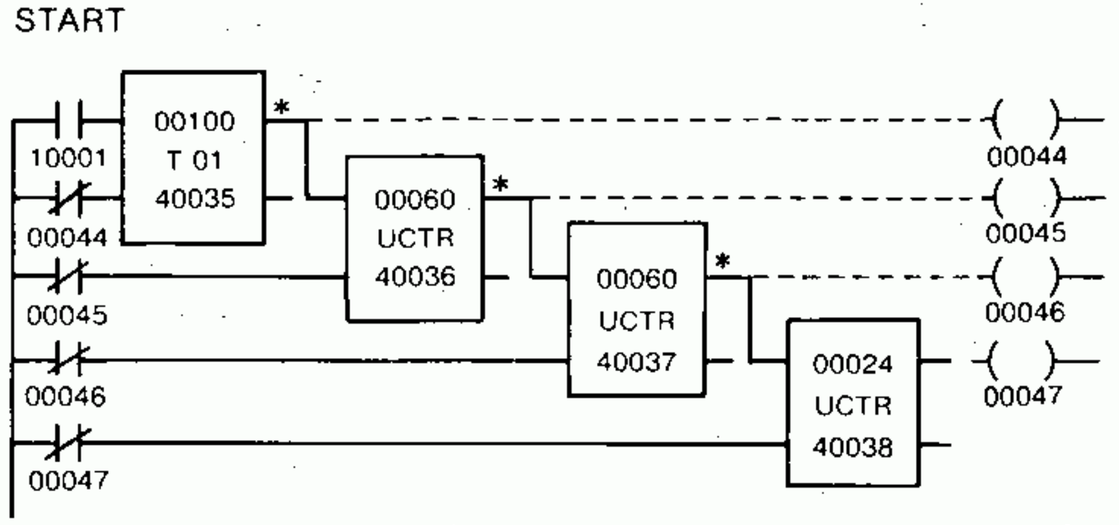


Fig. 5.26 Sample Timer Counter Cascaded Logic

- This clock operates while input relay 10001 is ON. The contents of 40036, 40037, and 40038 give the values in seconds, minutes, and hours, respectively. The clock is not provided with a compensating circuit.

(4) Circuit for Measuring Scan Time

- It is possible to make up a circuit for measuring the scanning time of the GL40S by combining an up counter with a 0.1-second timer.
- Fig. 5.27 shows a sample measuring circuit for scan time.

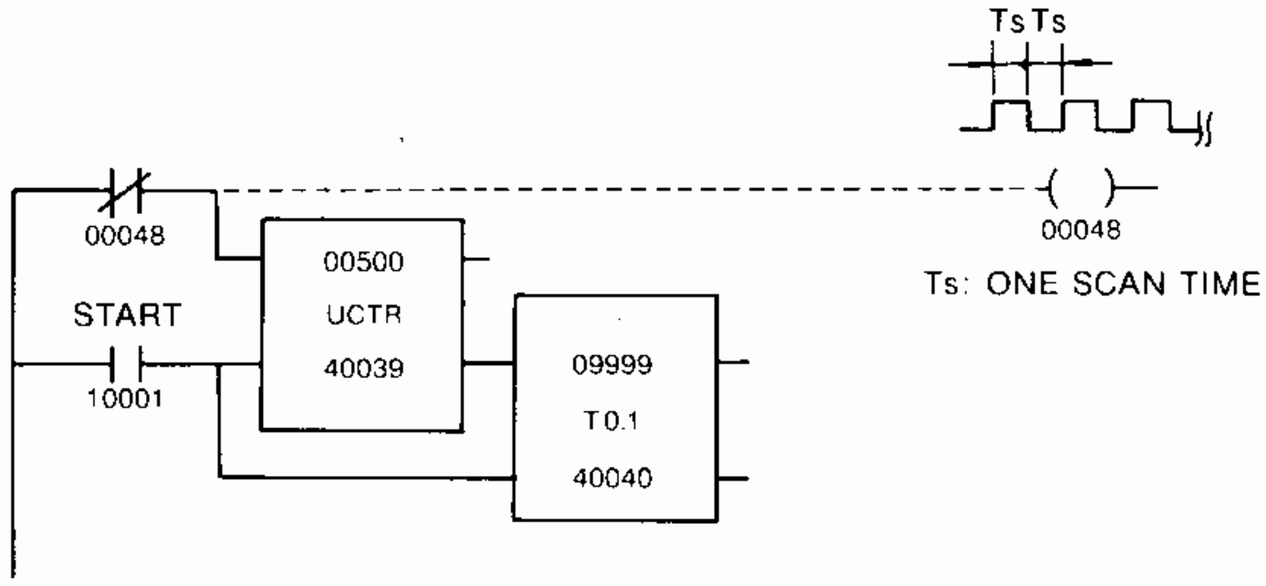


Fig. 5.27 A Circuit for Measuring Scan Time

- When input relay 10001 is turned ON, both the up counter and the 0.1-second timer start to measure the GL40S scanning time.
- Coil 00048 is turned ON and OFF alternately every scanning cycle. When the up counter counts 500 pulses (alternations of ON and OFF), the output 2 is turned OFF and the 0.1-second timer stops.
- Now the timer's current value (contents of 40040) is the time taken to count 500 pulses or, the time of 1000 scanning cycles given in units of 0.1 second.
- If, for example, the value is 205, then the average value of one scan time is $205 \times 0.1 \div 1000 = 20.5$ milli-seconds.

(5) Up/Down Counter

- An up/down counter can be made by combining an up counter with a down-counter.
- Fig. 5.28 shows an example of an up/down counter.

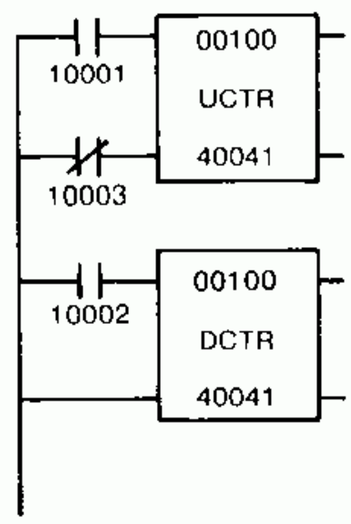


Fig. 5.28 Up/Down Counter

- The current value (contents of 40041) is increased by one each time input relay 10001 is turned from OFF to ON.
- The current value is decreased by one each time input relay 10002 is turned from OFF to ON.
- The current value is reset to zero when input relay 10003 is tuned ON.

5.5 ARITHMETIC FUNCTIONS

5.5.1 Types of Arithmetic Functions

The arithmetic operations are addition, subtraction, multiply, and divide applied on operand V_1 by operand V_2. There are eight variations of arithmetic functions as described in Table 5.15.

Table 5.15 Types of Arithmetic Functions

Type	Symbol	Operator	Range of Operand V_1	Range of Operand V_2	Reference Page
Addition	ADD	$V_1 + V_2$	0 – 9,999		80
Double-precision Addition	DADD		0 – 99,999,999		82
Subtraction	SUB	$V_1 - V_2$	0 – 9,999		84
Double-precision Subtraction	DSUB		0 – 99,999,999		86
Multiply	MUL	$V_2 \times V_2$	0 – 9,999		88
Double-precision Multiply	DMUL		0 – 99,999,999		90
Divide	DIV	$V_1 \div V_2$	0 – 9,999(0 – 99,989,999)*	0 – 9,999	91
Double-precision Divide	DDIV		0 – 9,999,999,899,999,999	0 – 99,999,999	95

*The range of V_1 becomes as shown in parentheses when two successive registers are used.

5.5.2 Addition (ADD)

(1) Function

Operates addition in 4-digit decimal without sign.

(2) Form

- Fig. 5.29 shows the form of addition (ADD)
- ADD is the symbol denoting the addition.
- Addition operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.16, specify any of constant K, reference number of various registers.

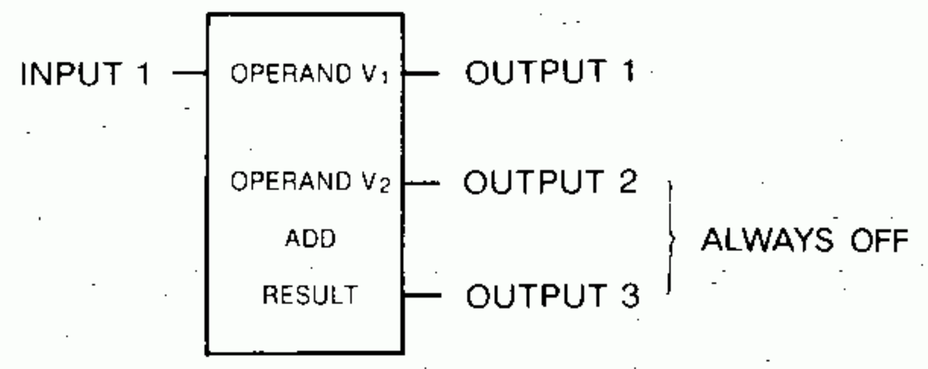


Fig. 5.29 ADD General Form

5.5 ARITHMETIC FUNCTIONS

5.5.1 Types of Arithmetic Functions

The arithmetic operations are addition, subtraction, multiply, and divide applied on operand V_1 by operand V_2. There are eight variations of arithmetic functions as described in Table 5.15.

Table 5.15 Types of Arithmetic Functions

Type	Symbol	Operator	Range of Operand V_1	Range of Operand V_2	Reference Page
Addition	ADD	$V_1 + V_2$	0 – 9,999		80
Double-precision Addition	DADD		0 – 99,999,999		82
Subtraction	SUB	$V_1 - V_2$	0 – 9,999		84
Double-precision Subtraction	DSUB		0 – 99,999,999		86
Multiply	MUL	$V_2 \times V_2$	0 – 9,999		88
Double-precision Multiply	DMUL		0 – 99,999,999		90
Divide	DIV	$V_1 \div V_2$	0 – 9,999(0 – 99,989,999)*	0 – 9,999	91
Double-precision Divide	DDIV		0 – 9,999,999,899,999,999	0 – 99,999,999	95

*The range of V_1 becomes as shown in parentheses when two successive registers are used.

5.5.2 Addition (ADD)

(1) Function

Operates addition in 4-digit decimal without sign.

(2) Form

- Fig. 5.29 shows the form of addition (ADD)
- ADD is the symbol denoting the addition.
- Addition operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.16, specify any of constant K, reference number of various registers.

Fig. 5.29 ADD General Form

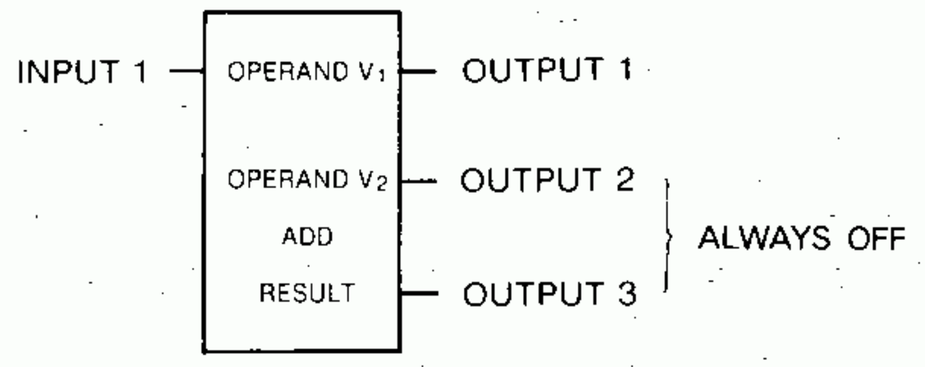


Table 5.16 Elements of ADD Function

Element Position	Specified Number	Description
Top	• Constant K (00000-09999) Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When constant K is specified, the value is the operand (V_1 = 0 to 9,999). • When register reference Nos. are specified, the contents are the operand (V_1 = 0 to 9,999).
Middle		• When constant K is specified, the value is the operand (V_2 = 0 to 9999). • When register reference Nos. are specified, the contents are the operand (V_2 = 0 to 9,999).
Bottom	• Holding register (40001-42048) • Link register (R0001-R1024)	The result of addition function (0 to 9,999) is stored in 4×××× or R××××.

(3) Operation

- By the addition (ADD), $V_1 + V_2$ is calculated when the input 1 is ON. The result is treated as follows.

  (a) If $V_1 + V_2 \leqq 9{,}999$,

  $V_1 + V_2$ is stored in R. The Output 1 remains OFF.

  (b) If $V_1 + V_2 \geqq 10{,}000$, $V_1 + V_2 - 10{,}000$ is stored in R. The output 1 is turned ON.

- The outputs 2 and 3 are always OFF.
- The result remains in R even after the input 1 is turned from ON to OFF.
- Table 5.17 shows an ADD operation.

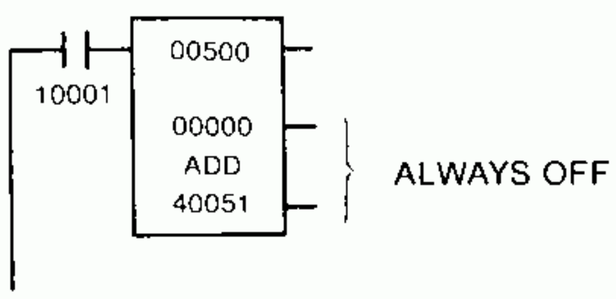
Table 5.17 ADD Operation

Input 1	Condition	Operation	Output 1
ON	$V_1 + V_2 \leqq 9{,}999$	$V_1 + V_2 \rightarrow R$	OFF
	$V_1 + V_2 \geqq 10{,}000$	$V_1 + V_2 - 10{,}000 \rightarrow R$	ON
OFF	None	Not operated.	OFF

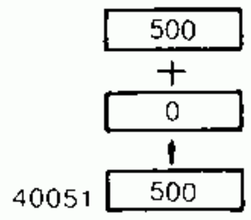
Note $V_1 + V_2 \rightarrow R$ indicates that $V_1 + V_2$ operation result is stored in R.

(4) Example

Example 1:



(a) Ladder

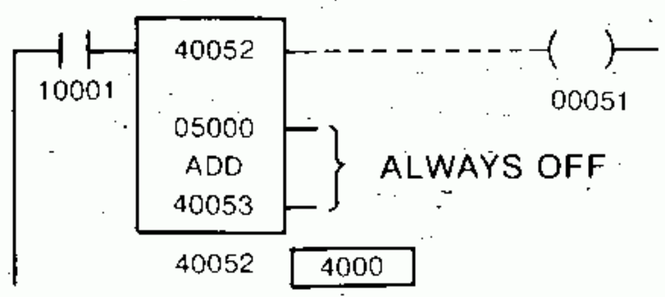


(b) ADD Operation

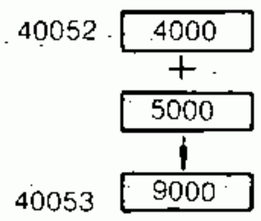
ADD in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 remains OFF. The result remains in 40051 even after input relay 10001 is turned OFF.

5.5.2 Addition (ADD) (Cont'd)

Example 2:



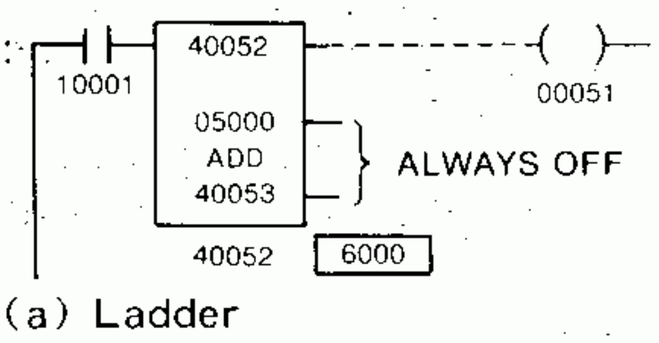
(a) Ladder



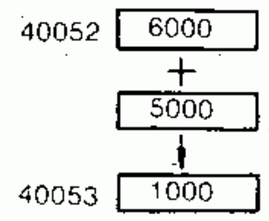
(b) ADD Operation

ADD in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 remains OFF. The result remains in 40053 even after input relay 10001 is turned **OFF.**

Example 3:



(a) Ladder



(b) ADD Operation

ADD in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 (coil 00051) is turned ON. The result remains in 40053 even after input relay 10001 is turned OFF.

5.5.3 Double-precision Addition (DADD)

(1) Function

Operates double-precision addition in 8-digit decimal without sign.

(2) Form

- Fig. 5.30 shows the form of double-precision addition (DADD).

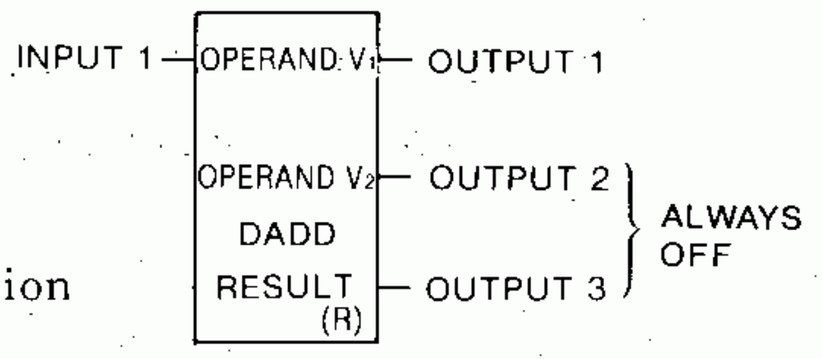


Fig. 5.30 DADD General Form

- DADD is the symbol denoting the double-precision addition.
- Double-precision addition operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.18, specify any reference number of various registers.

Table 5.18 Elements of DADD Function

Element Position	Specified-Number	Description
Top	Either one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1023)	Operand (V_1 = 0 to 99,999,999) is stored as follows. ××××× → V_1H ×××××+1 → V_1L V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle		Operand (V_2 = 0 to 99,999,999) is stored as follows. ××××× → V_2H ×××××+1 → V_2L V_2H: Higher-place 4 digits of V_2 V_2L: Lower-place 4 digits of V_2
Bottom	• Holding register (40001-42048) • Link register (R0001-R1023)	Result of operation ($V_1 + V_2$ = 0 to 99,999,999) is stored as follows. ××××× → $(V_1+V_2)H$ ×××××+1 → $(V_1+V_2)L$ $(V_1+V_2)H$: Higher-place 4 digits of (V_1+V_2) $(V_1+V_2)L$: Lower-place 4 digits of (V_1+V_2)

(3) Operation

- By DADD, $V_1 + V_2$ is calculated when the input 1 is ON. The result is treated as follows.

  (a) If $V_1 + V_2 \leqq 99,999,999$,

  The four higher-place digits of $V_1 + V_2$ are stored in R and the four lower-place digits in R + 1. The output 1 remains OFF.

  (b) If $V_1 + V_2 \geqq 100,000,000$,

  The four higher-place digits of $V_1 + V_2 - 100,000,000$ are stored in R and the four lower-place digits in R + 1. The output 1 is turned ON.

- The outputs 2 and 3 are always OFF.
- The result remains in R and R + 1 even after the input 1 is turned from ON to OFF.
- Table 5.19 shows a DADD.

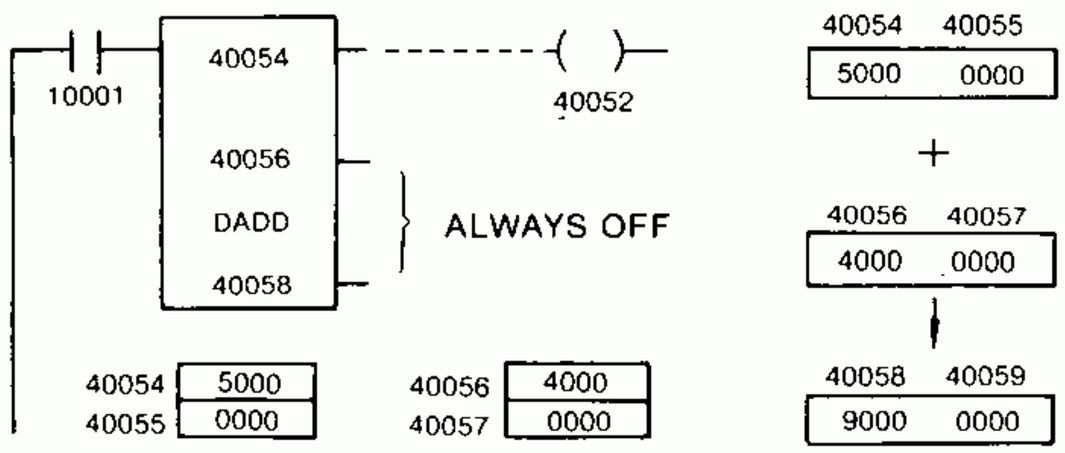
Table 5.19 Double-precision Addition Operation

Input 1	Condition	Operation	Output 1
ON	$V_1 + V_2 \leqq 99,999,999$	$V_1 + V_2 \rightarrow$ R(Higher-place 4 digits), R + 1 (Lower-place 4 digits).	OFF
ON	$V_1 + V_2 \geqq 100,000,000$	$V_1 + V_2 - 100,000,000 \rightarrow$ R (Higher-place 4 digits), R + 1 (Lower-place 4 digits).	ON
OFF	None	Not operated.	OFF

Note $V_1 + V_2 \rightarrow$ R and R 1 indicate that $V_1 + V_2$ operation result is stored in R and R + 1, respectively.

(4) Example

Example 1:

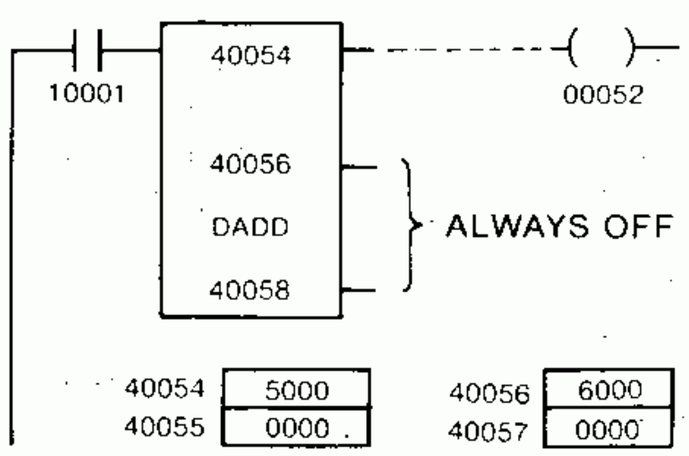


(a) Ladder (b) DADD Operation

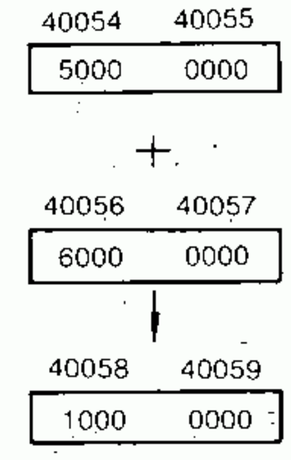
DADD in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 remains OFF. The results remain in 40058 and 40059 even after input relay 10001 is turned OFF.

5.5.3 Double-precision Addition (DADD) (Cont'd)

Example 2:



(a) Ladder



(b) DADD Operation

DADD in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 (coil 00052) is turned ON. The result remains in 40058 and 40059 even after input relay 10001 is turned OFF.

5.5.4 Subtraction (SUB)

(1) Function

Operates subtraction in 4-digit decimal without sign.

(2) Form

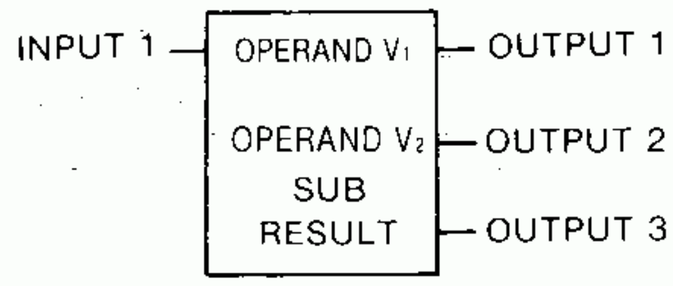


Fig. 5.31 SUB General Form

- Fig. 5.31 shows the form of subtraction (SUB).
- Subtraction operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.20, specify any of constant K or the reference number of various registers.

Table 5.20 Elements of SUB Function

Element Position	Specified Number	Description
Top	• Constant K (00000-09999) Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When constant K is specified, the value is the operand (V_1 = 0 to 9,999). • When register reference Nos. are specified, the contents are the operand (V_1 = 0 to 9,999).
Middle		• When contant K is specified, the value is the operand (V_2 = 0 to 9,999) • When register reference Nos. are specified, the contents are the operand (V_2 = 0 to 9,999).
Bottom	• Holding register (40001-42048) • Link register (R0001-R1024)	The result of subtraction fuction 0 to 9,999) is stored in 4×××× or R××××.

(3) Operation

- By the subtraction (SUB), V_1 - V_2 will be calculated when the input 1 is ON. The result is treated as follows.

  (a) If $V_1 > V_2$

  V_1 - V_2 is stored in R and only the output 1 is turned ON.

  (b) If $V_1 = V_2$,

  V_1 - V_2 = 0 is stored in R and only the output 2 is turned ON.

  (c) If $V_1 < V_2$,

  V_2 - V_1 is stored in R and only the output 3 is turned ON.

- The result remains in R even after the input 1 is turned from ON to OFF.
- Table 5.21 shows a SUB operation.

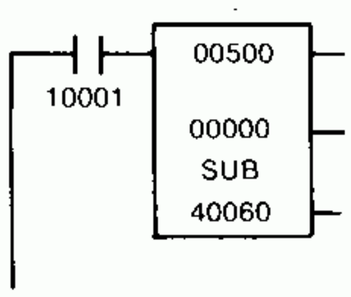
Table 5.21 SUB Operation

Input 1	Condition	Operation	Output 1	Output 2	Output 3
ON	$V_1 > V_2$	$V_1 - V_2 \rightarrow R$	**ON**	OFF	OFF
	$V_1 = V_2$	$0 \rightarrow R$	OFF	**ON**	OFF
	$V_1 < V_2$	$V_2 - V_1 \rightarrow R$	OFF	OFF	**ON**
OFF	None	Not operated.	OFF	OFF	OFF

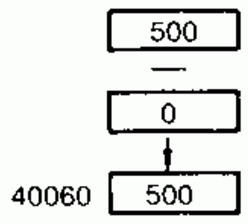
Note $V_1 - V_2 \rightarrow R$ indicates that $V_1 - V_2$ operation result is stored in R.

(4) Example

Example 1:



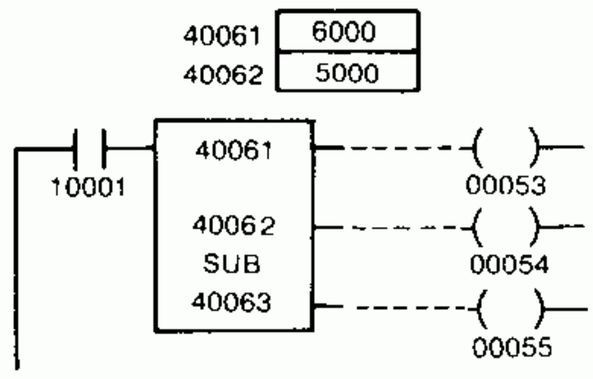
(a) Ladder



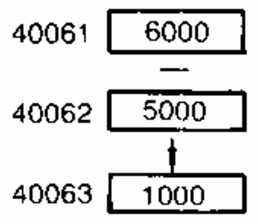
(b) SUB Operation

SUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned ON. The result remains in 40060 even after input relay 10001 is turned OFF.

Example 2:



(a) Ladder

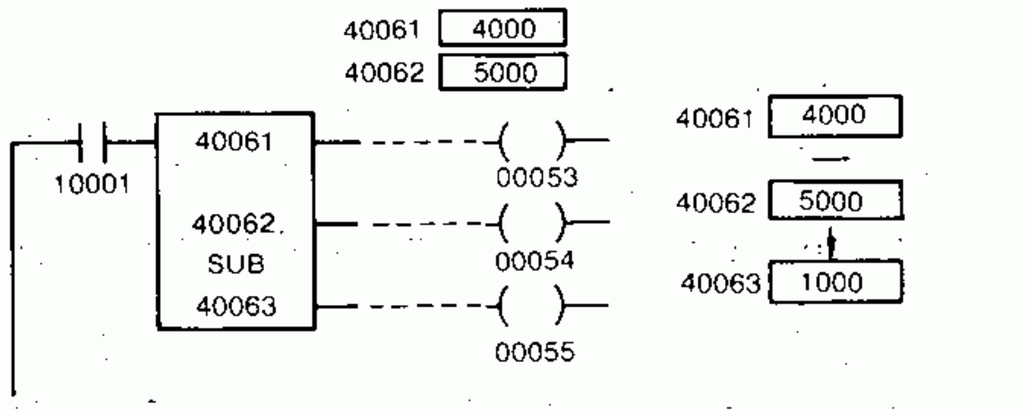


(b) SUB Operation

SUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 (coil 00053) is turned ON. The result remains in 40063 even after input relay 10001 is turned OFF.

5.5.4 Subtraction (SUB) (Cont'd)

Example 3:



(a) Ladder (b) SUB Operation

SUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 3 (coil 00055) is turned ON. The result remains in 40063 even after input relay 10001 is turned OFF.

5.5.5 Double-precision Subtraction (DSUB)

(1) Function

Operates double-presion subtraction in 8-digit decimal without sign.

(2) Form

- Fig. 5.32 shows the form of double-precision subtraction (DSUB).

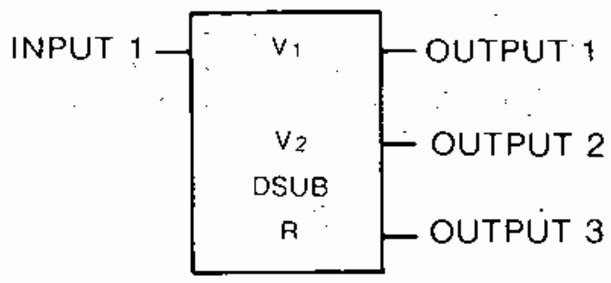


Fig. 5.32 DSUB General Form

- DSUB is the symbol denoting the double-precision subtraction.
- Double-precision subtraction operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.22, specify any reference number of various registers.

Table 5.22 Elements of DSUB Function

Element Position	Specified Number	Description
Top	Either one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1023)	Operand(V_1 = 0 to 99,999,999) is stored as follows. ×××××　　××××× + 1 [V_1H]　　[V_1L] V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle		Operand(V_2 = 0 to 99,999,999) is stored as follows. ×××××　　××××× + 1 [V_2H]　　[V_2L] V_2H: Higher-place 4 digits of V_2 V_2L: Lower-place 4 digits of V_2-V
Bottom	• Holding register (40001-42048) • Link register (R0001-R1023)	Result of operation($\lvert V_1 - V_2 \rvert$ =0 to 99,999,999) is stored as follows. ×××××　　××××× + 1 [$\lvert V_1 - V_2 \rvert H$]　　[$\lvert V - V_2 \rvert L$] $\lvert V_1 - V_2 \rvert H$: Higher-place 4 digits of $\lvert V_1 - V_2 \rvert H$ $\lvert V_1 - V_2 \rvert L$: Lower-place 4 digits of $\lvert V_1 - V_2 \rvert L$

(3) Operation

- By the double-precision subtraction (DSUB), V_1 - V_2 is calculated when the input 1 is ON. The result is treated as follows.

  (a) If $V_1 > V_2$,

  The four higher-place digits of V_1 - V_2 are stored in R and the four lower-place digits in R+1. Only the output 1 is turned ON.

  (b) If V_1 - V_2,

  $V_1 - V_2 = 0$ is stored in R and R+1 and only the output 2 is turned ON.

  (c) If $V_1 < V_2$,

  The four higher-place digits of V_2 - V_1 are stored in R and the four lower-place digits in R+1. Only the output 3 is turned ON.

- The result remains in R and R+1 even after the input 1 is turned from ON to OFF.
- Table 5.23 shows a double-precision subtraction operation.

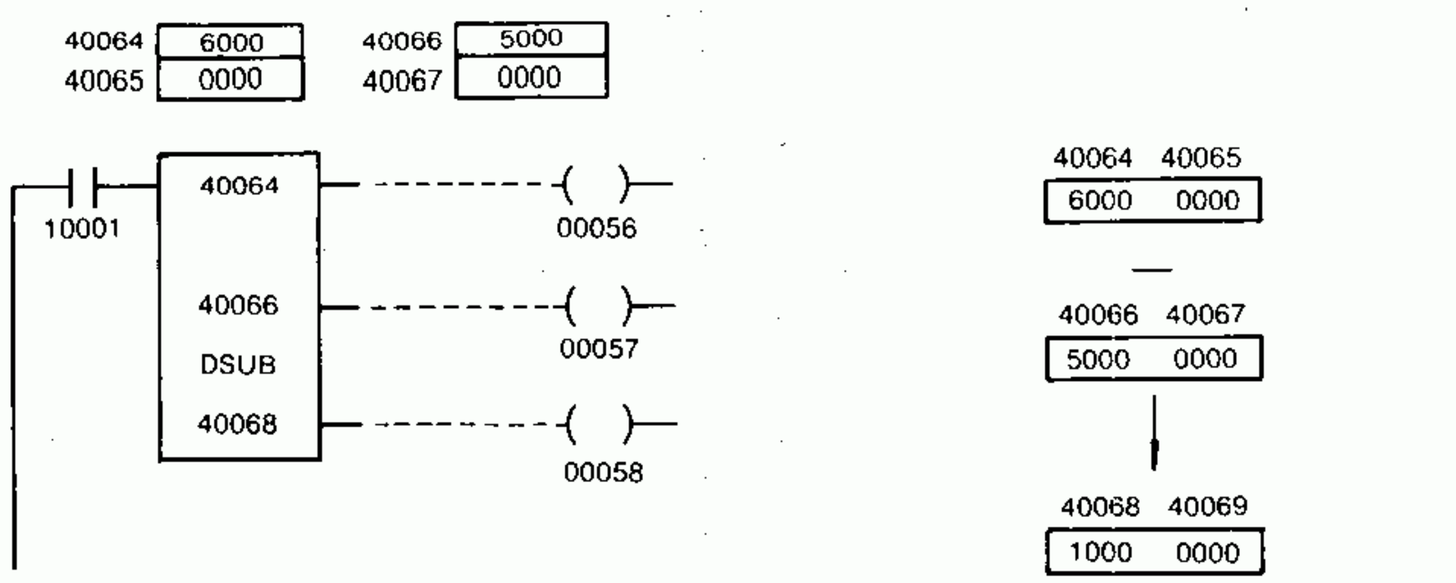
Table 5.23 DSUB Operation

Input 1	Condition	Operation	Output 1	Output 2	Output 3
ON	$V_1 > V_2$	$V_1 - V_2 \rightarrow$R (Higher-place 4 digits), R+1 (Lower-place 4 digits).	ON	OFF	OFF
	$V_1 = V_2$	$0 \rightarrow$R, R+1	OFF	ON	OFF
	$V_1 < V_2$	$V_2 - V_1 \rightarrow$R (Higher-place 4 digits), R+1 (Lower-place 4 digits).	OFF	OFF	ON
OFF	None	Not operated.	OFF	OFF	OFF

Note V_1-$V_2 \rightarrow$R and R+1 indicates that V_1 - V_2 operation result is stored in R and R+1, respectively.

(4) Example

Example 1:

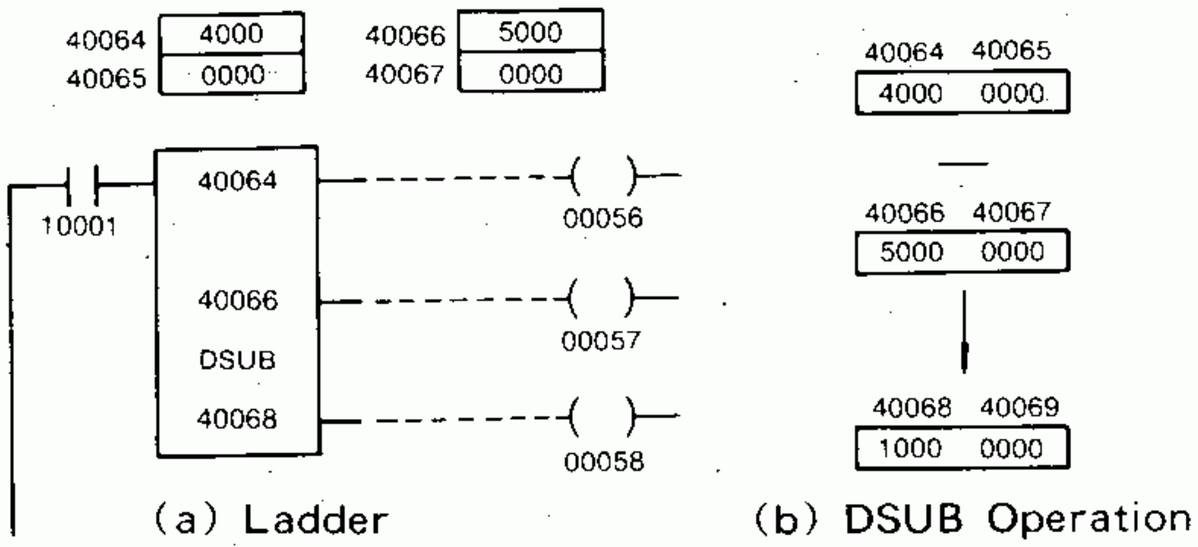


(a) Ladder (b) DSUB Operation

DSUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 (coil 00056) is turned ON. The result remains in 40068 and 40069 even after input relay 10001 is turned OFF.

5.5.5 Double-precision Subtraction (DSUB) (Cont'd)

Example 2:



(a) Ladder (b) DSUB Operation

DSUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 3 (coil 00058) is turned ON. The result remains in 40068 and 40069 even after input relay 10001 is turned OFF.

5.5.6 Multiply (MUL)

(1) Function

Operates multiply in 4-digit decimal without sign.

(2) Form

- Fig. 5.33 shows the form of multiply (MUL).

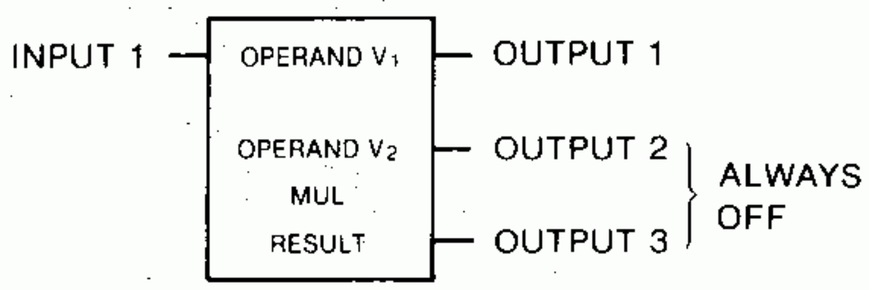


Fig. 5.33 MUL General Form

- MUL is the symbol denoting the multiply.
- Multiply operation requires three elements placed vertically (top, middle, and bottom) Referring to Table 5.24, specify any of constant K or the reference number of various registers.

Table 5.24 Elements of MUL Function

Element Position	Specified Number	Description
Top	Constant K (00000-09999) Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When constant K is specified, the value is the operand (V_1 = 0 to 9,999). • When register reference Nos. are specified, the contents are the operand (V_2 = 0 to 9,999).
Middle		• When constant K is specified, the value is the operand (V_2 = 0 to 9,999). • When register reference Nos. are specified, the contents are the operand (V_2 = 0 to 9,999).
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of operation ($\lvert V_1+V_2 \rvert$ = 0 to 99,999,999) is stored as follows. ×××××　　××××× + 1 [$(V_1+V_2)H$]　[$(V+V_2)L$] $(V_1+V_2)H$: Higher-place 4 digits of $(V_1+V_2)H$. $(V_1+V_2)L$: Lower-place 4 digits of $(V_1+V_2)L$.

(3) Operation

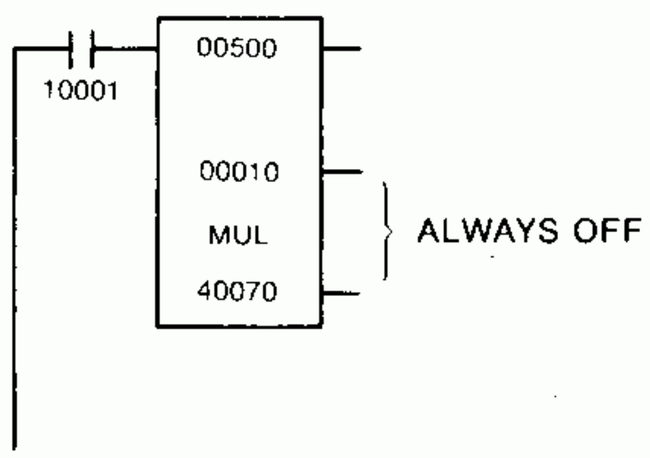
- By the multiply (MUL), $V_1 \times V_2$ is calculated when the input 1 is ON. The result is treated as follows. The four higher-place digits of $V_1 \times V_2$ are stored in R and the four lower-place digits in R+1. Output 1 is turned ON.
- Table 5.25 shows a MUL operation.

Table 5.25 MUL Operation

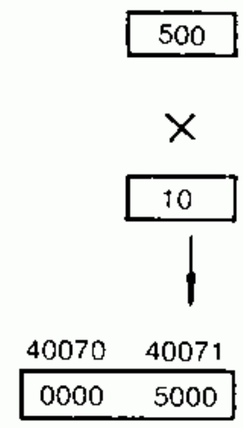
Input 1	Condition	Operation	Output 1
ON	None	$V_1 \times V_2 \rightarrow$ R (Higher-place 4 digits), R+1 (Lower-place 4 digits).	ON
OFF	None	Not operated.	OFF

(4) Example

Example 1:



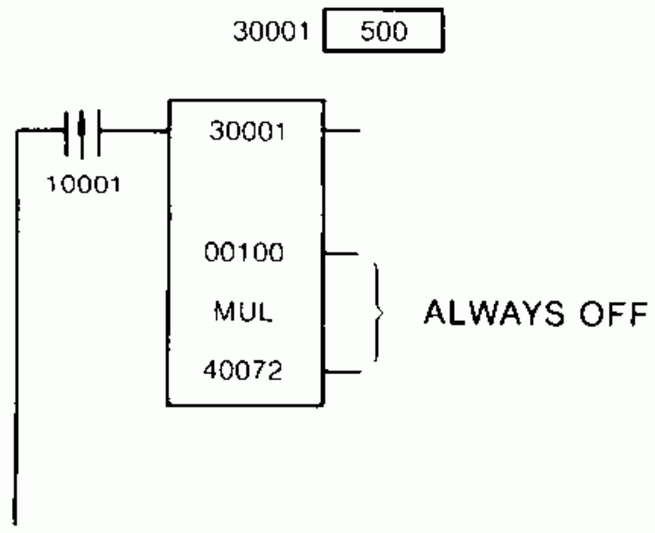
(a) Ladder



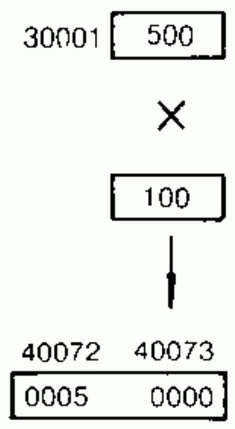
(b) MUL Operation

MUL in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 is turned ON. The result remains in 40070 and 40071 even after input relay 10001 is turned OFF.

Example 2:



(a) Ladder



(b) MUL Operation

MUL in (a) executes the operation of (b) only during the scanning cycle when input relay 10001 is turned ON. The output 1 is turned ON. The result remains in 40072 and 40073 even after input relay 10001 is turned OFF.

5.5.7 Double-precision Multiply (DMUL)

(1) Function

Operates double-precision multiply in 8-digit decimal without sign.

(2) Form

- Fig. 5.34 shows the form of double-precision multiply (DMUL).

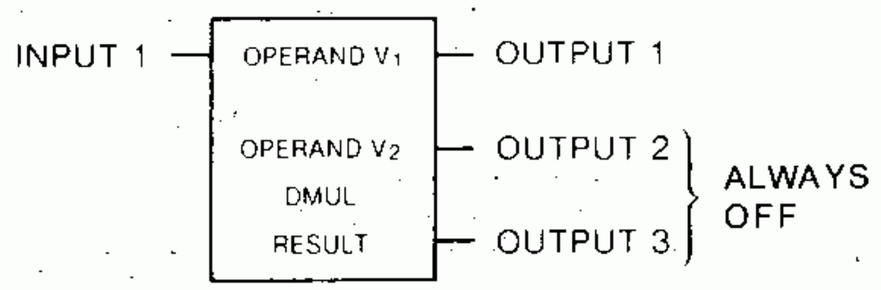


Fig. 5.34 DMUL General Form

- DMUL is the symbol denoting the double-precision multiply.
- Double-precision multiply operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.26, specify any reference.

Table 5.26 Elements of DMUL Function

Element Position	Specified Number	Description
Top	Either one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1023)	Operand (V_1 = 0 to 99,999,999) is stored as follows. ×××× (+1) [V_1H] [V_1L] V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle		Operand (V_2 = 0 99,999,999) is stored as follows. ×××× (+1) [V_2H] [V_2L] V_2H: Higher-place 4 digits of V_2 V_2L: Lower-place 4 digits of V_2
Bottom	• Holding register (40001-42045) • Link register (R0001-R1021)	Result of operation ($V_1 \times V_2$ 0 to 99,999,999) is stored as follows. ×××× (+1) (+2) (+3) [$(V_1 \times V_2) H_1$] [$(V_1 \times V_2) H_2$] [$(V_1 \times V_2) L_1$] [$(V_1 \times V_2) L_2$] $(V_1 \times V_2) H_1$: Most significant 4 digits of $(V_1 \times V_2)$ $(V_1 \times V_2) H_2$: Higher-place 4 digits of $(V_1 \times V_2)$ $(V_1 \times V_2) L_1$: Lower-place 4 digits of $(V_1 \times V_2)$ $(V_1 \times V_2) L_2$: Least significant 4 digits of $(V_1 \times V_2)$

(3) Operation

- By the double-precision multiply (DMUL), $V_1 \times V_2$ is calculated when the input 1 is ON. The result is treated as follows.

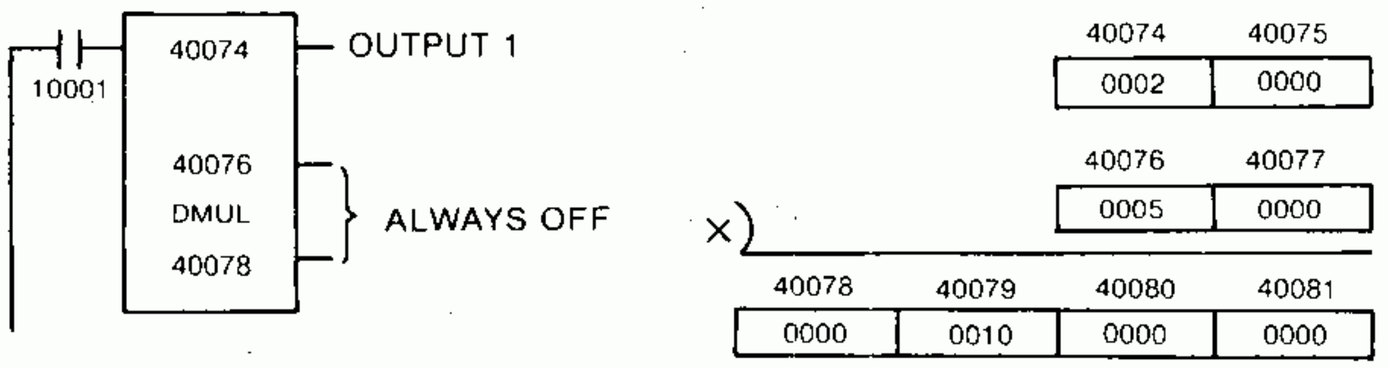
  The most significant four digits of $V_1 \times V_2$ are stored in R, the second most significant four digits in R+1, the second least significant four digits in R+2, the least significant four digits in R+3, and the output 1 is turned ON.

- The result remains in R, R+1, R+2, and R+3 after the input 1 is turned from ON to OFF.
- Table 5.27 shows a DMUL multiply.

Table 5.27 DMUL Operation

Input 1	Condition	Operation	Output 1
ON	None	$V_1 \times V_2 \rightarrow R$ (Most significant 4 digits), R+1 (2nd most significant 4 digits), R+2 (2nd least significant 4 digits), R+3 (Least significant 4 digits).	ON
OFF	None	Not operated.	OFF

(4) Example



(a) Ladder (b) DMUL Operation

DMUL in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 is turned ON. The result remains in 40078-40081 even after input relay 10001 is turned OFF.

5.5.8 Divide (DIV)

(1) Function

Operates divide in 4-digit decimal without sign.

(2) Form

- Fig. 5.35 shows the form of divide (DIV).

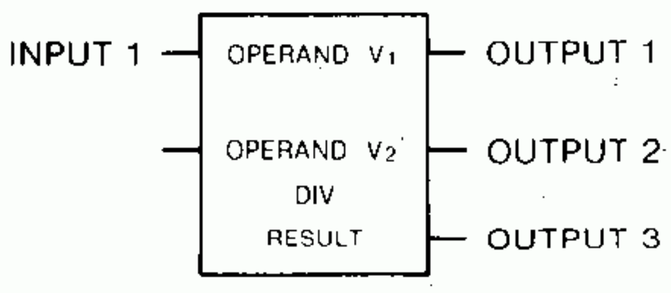


Fig. 5.35 DIV General Form

- DIV is the symbol denoting the divide.
- Divide operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.28, specify any of constant K or reference number of various registers.

Table 5.28 Elements of DIV Function

Element Position	Specified Number	Description
Top	• Constant K (00000-09999) Any one of the following: • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	• When constant K is specified, the value is the operand (V_1=0 to 9,999). • When register reference Nos. are specified, the operand (V_1=0 to 99,989,999) is stored as follows. ××××× ××××× +1 [V_1H] [V_1L] V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle	• Constant K (00000-09999) Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When constant K is specified, the value is the operand (V_2=1 to 9,999). • When register reference Nos. are specified, the contents are the operand (V_2=1 to 9,999).
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of divide function ($V_1 \div V_2$) is stored as follows. Where input 2 is OFF, Where input 2 is ON. ××××× ××××× +1 ××××× ××××× +1 [ ] [ ] [ ] [ ] The quotient / The remainder / The integer quotient / The decimal quotient

(3) Operation

• By the divide (DIV), $V_1 \div V_2$ is calculated when the input 1 is ON. The result is treated as follows.

① If the input 2 is OFF,

The quotient of $V_1 \div V_2$ is stored in R and the remainder in R+1. Only the output 1 is turned ON.

② If the input 2 is ON,

The integer part of the quotient of $V_1 \div V_2$ is stored in R and the decimal part (rounded off to the fifth decimal place) in R+1. Only the output 1 is turned ON.

③ The result remains in R and R+1 even after the input 1 is turned from ON to OFF.

④ In the following cases, divide operation is not executed and zero is placed in each of R and R+1.

(Ⅰ) When $V_2 = 0$, the output 3 is turned ON.

(Ⅱ) If the quotient or the integer part of quotient overflows in R, the output 2 is turned ON.

• Tables 5.29 and 5.30 show a divide operation (DIV).

Table 5.29 DIV Operation (Constant in Top Place)

Input 1	Input 2	Condition	Operation	Output 1	Output 2	Output 3
ON	OFF	$V_2 \neq 0$	$V_1 \div V_2 \rightarrow \begin{matrix} R & \text{(Quotient)} \\ R+1 & \text{(Remainder)} \end{matrix}$	ON	OFF	OFF
		$V_2 = 0$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	OFF	ON
ON	ON	$V_2 \neq 0$	$V_1 \div V_2 \rightarrow \begin{matrix} R & \text{(Integer quotient)} \\ R+1 & \text{(Decimal quotient)} \end{matrix}$	ON	OFF	OFF
		$V_2 = 0$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	OFF	ON
OFF	ON OFF	None	Not operated.	OFF	OFF	OFF

NOTE $V_1 \div V_2 \rightarrow R$ and $R + 1$ indicate that a result of $V_1 \div V_2$ is stored R and R + 1 respectively .

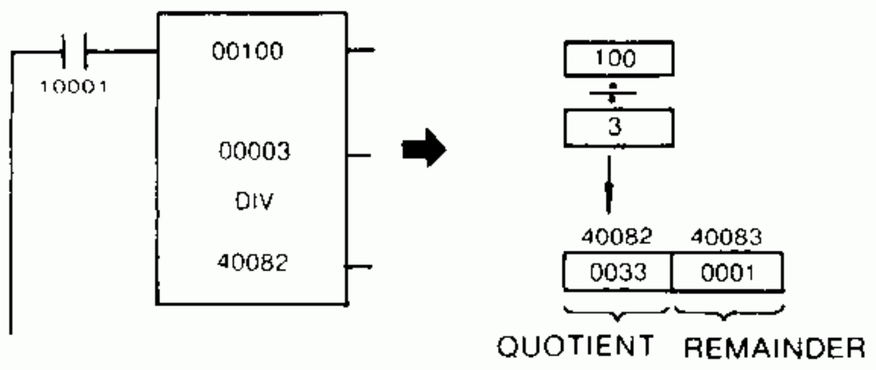
Table 5.30 DIV Operation (Register in Top Place)

Input 1	Input 2	Condition	Operation	Output 1	Output 2	Output 3
ON	OFF	$V_2 \neq 0,\ V_{1H} < V_2$	$(V_{1H} \times 10000 + V_{1L}) \div V_2 \rightarrow \begin{matrix} R & \text{(Quotient)} \\ R+1 & \text{(Remainder)} \end{matrix}$	ON	OFF	OFF
		$V_2 \neq 0,\ V_{1H} \geqq V_2$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	ON	OFF
		$V_2 = 0$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	OFF	ON
ON	ON	$V_2 \neq 0,\ V_{1H} < V_2$	$(V_{1H} \times 10000 + V_{1L}) \div V_2 \rightarrow \begin{matrix} R & \text{(Integer quotient)} \\ R+1 & \text{(Decimal quotient)} \end{matrix}$	ON	OFF	OFF
		$V_2 \neq 0,\ V_{1H} \geqq V_2$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	ON	OFF
		$V_2 = 0$	$0 \rightarrow \begin{matrix} R \\ R+1 \end{matrix}$	OFF	OFF	ON
OFF	ON OFF	None	Not operated.	OFF	OFF	OFF

Note $V_1 \div V_2 \rightarrow R$ and $R + 1$ indicate that a result of $V_1 \div V_2$ is stored R and R + 1 respectively.

(4) Example–Divide

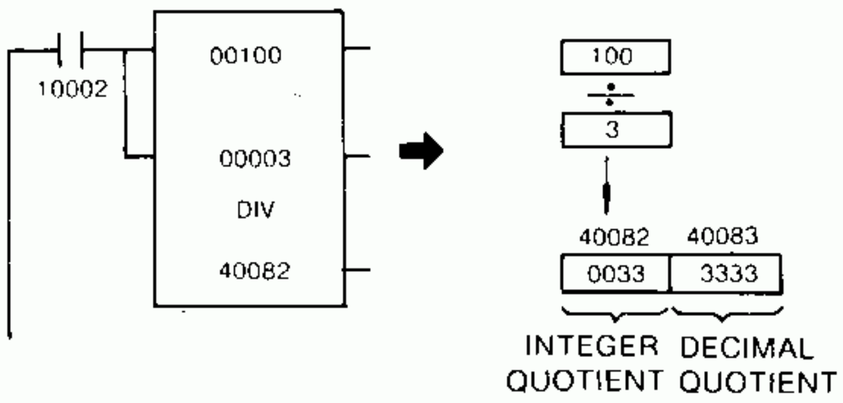
Example 1:



(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned ON. The result remains in 40082 and 40083 even after input relay 10001 is turned OFF.

Example 2:

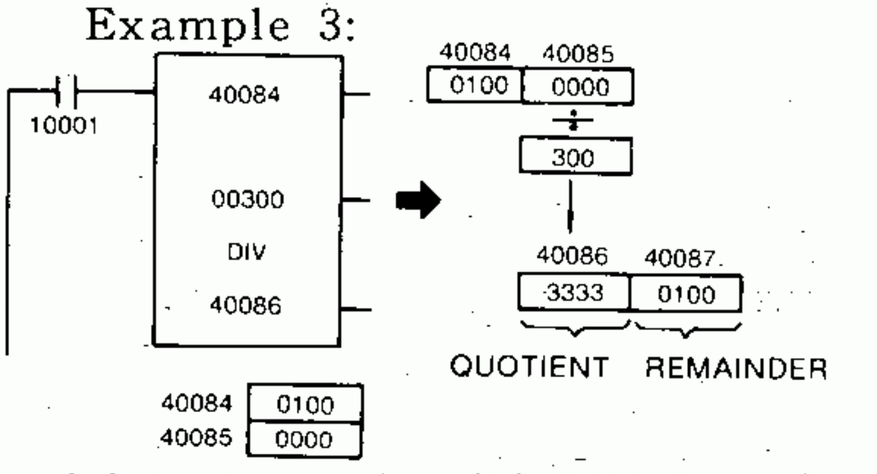


(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10002 is ON. Only the output 1 is turned ON. The result remains in 40082 and 40083 even after input relay 10002 is turned OFF.

5.5.8 Divide (DIV) (Cont'd)

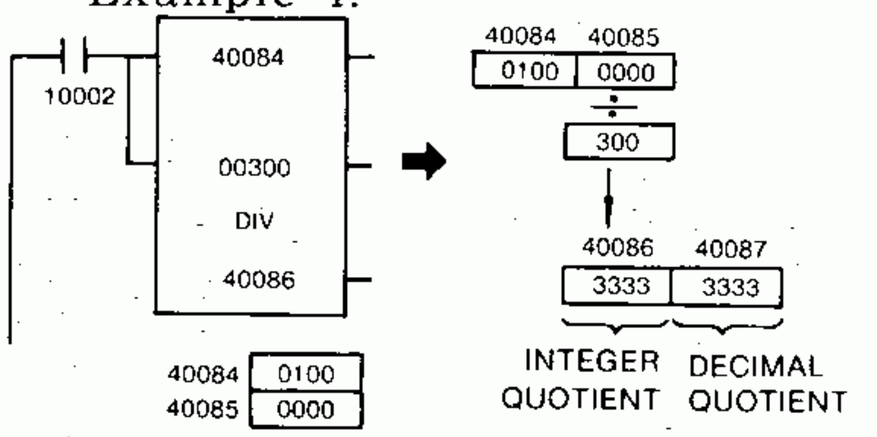
Example 3:



(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned ON. The result remains in 40086 and 40087 even after input relay 10001 is turned OFF.

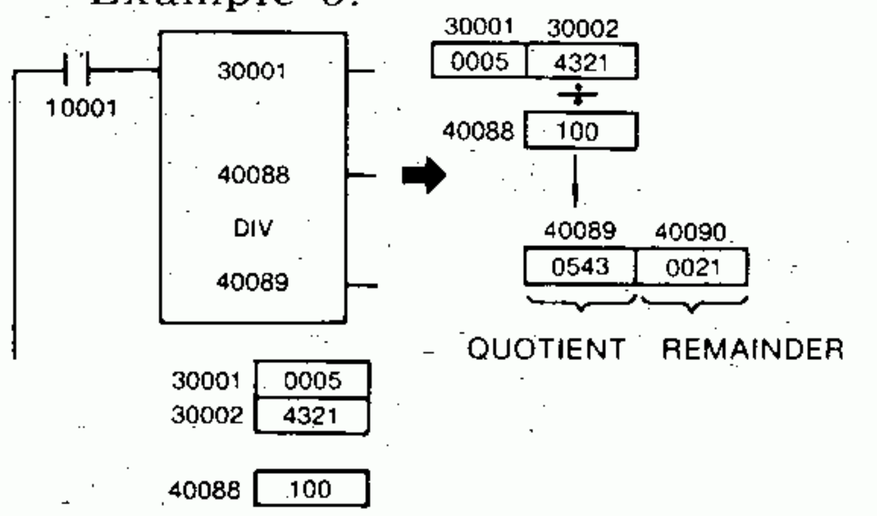
Example 4:



(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10002 is ON. Only the output 1 is turned ON. The result remains in 40086 and 40087 even after input relay 10002 is turned OFF.

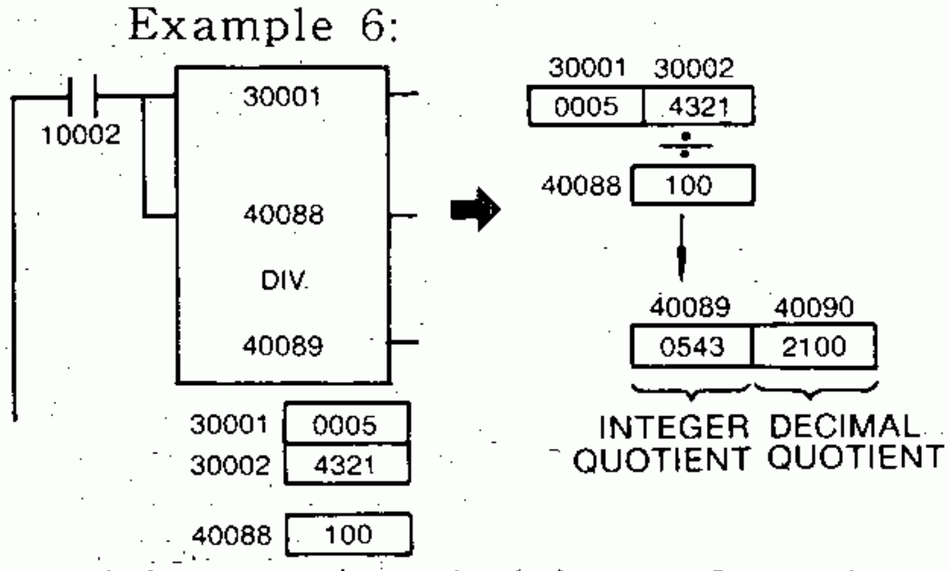
Example 5:



(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned ON. The result remains in 40089 and 40090 even after input relay 10001 is turned OFF.

Example 6:



(a) Ladder (b) DIV Operation

DIV in (a) executes the operation of (b) when input relay 10002 is ON. Only the output 1 is turned ON. The result remains in 40089 and 40090 even after input relay 10002 is turned OFF.

5.5.9 Double-precision Divide Function (DDIV)

(1) Function

Operates double-precision divide function in 8-digit decimal without sign.

(2) Form

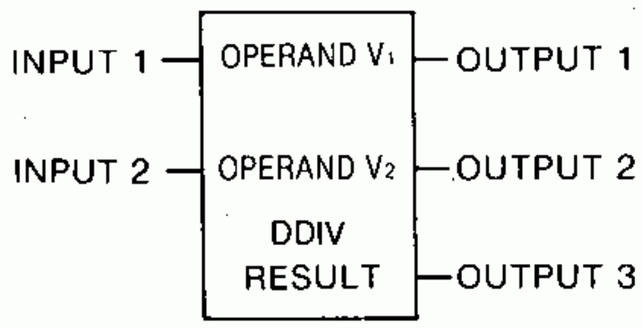


Fig. 5.36 DDIV General Form

- Fig. 5.36 shows the form of double-presision divide (DDIV).
- DDIV is the symbol denoting the double-precision divide.
- Double-precision divide operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.31, specify any reference number of various registers.

Table 5.31 Elements of DDIV Function

Element Position	Pecified Number	Description
Top	Either one of the following: • Input register (30001-30125) • Holding register (40001-42045) • Link register (R0001-R1021)	The operand (V_1=0 to 9,999,999,899,999,999) is stored as follows. ×××××: V_1H_1 ×××××+1: V_1H_2 ×××××+2: V_1L_1 ×××××+3: V_1L_2 V_1H_1: Most significant 4 digits of V_1 V_1H_2: 2nd most significant 4 digits of V_1 V_1L_1: 2nd least significant 4 digits of V_1 V_1L_2: Least significant 4 digits of V_1
Middle	Either one of the following: • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-1023)	The operand (V_2=1 to 99,999,999) is stored as follows. ×××××: V_2H ×××××+1: V_2L V_2H: Higher-place 4 digits of V_2 V_2L: Lower-place 4 digits of V_2
Bottom	Either one of the following: • Holding register (40001-42045) • Link register (R0001-R1021)	The result of operation ($V_1 \div V_2$) is stored as follows. Where input 2 is OFF, ×××××: Quotient of ($V_1 \div V_2$) ×××××+1: Remainder of ($V_1 \div V_2$) Where input 2 is ON, ×××××: Integer quotient of ($V_1 \div V_2$) ×××××+1: Decimal quotient of ($V_1 \div V_2$)

(3) Operation

By the double-precision divide (DDIV), $V_1 \div V_2$ is calculated when the input 1 is ON. The result is treated as follows.

① If the input 2 is OFF,

- The four higher-place digits of the quotient of $V_1 \div V_2$ are stored in R and the four lower-place digits in R+1.
- The four higher-place digits of the remainder of $V_1 \div V_2$ are stored in R+2 and the four lower-place digits in R+3.
- Only the output 1 is turned ON.

② If the input 2 is ON,

- The four higher-place digits of the integer part of the quotient of $V_1 \div V_2$ are stored in R and the four lower-place digits in R+1.
- The four high-place digits of the decimal part of the quotient of $V_1 \div V_2$ are stored in R+2 and the four lower-place digits in R+3.
- Only the output 1 is turned ON.

③ The result remains in each of R, R+1, R+2, and R+3 even after the input 1 is turned from ON to OFF.

④ In the following cases, the divide operation is not executed and zero is placed in each of R, R+1, R+2, and R+3.

(Ⅰ) When $V_2 = 0$, the output 3 is turned ON.

(Ⅱ) If the quotient or the integer part of quotient overflows in R and R+1, the output 2 is turned ON.

Example: Where $V_1 = 5{,}000{,}000{,}000$, and $V_2 = 10$, the quotient 5,000,000,000 cannot be stored in R and R+1.

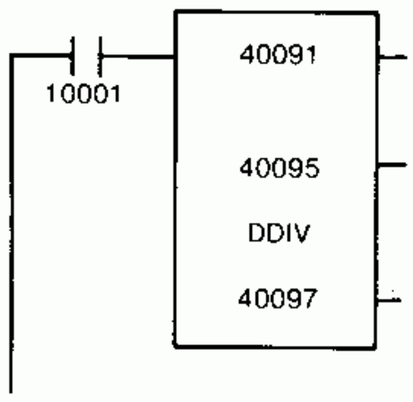
Table 5.32 shows a double-precision divide operation (DDIV).

Table 5.32 DDIV Operation

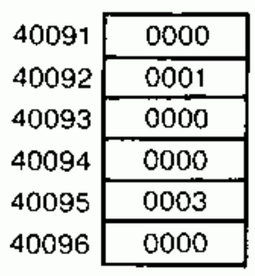
Input 1	Input 2	Condition	Operation	Output 1	Output 2	Output 3
ON	OFF	$V_2 = (V_2H \times 10^4 + V_2L) \neq 0$, $(\underbrace{V_1H_1 \times 10^4 + V_1H_2}_{\text{Higher-place 8 digits of } V_1}) < V_2$	$(\underbrace{V_1H_1 \times 10^{12} + V_1H_2 \times 10^8 + V_1L_1 \times 10^4 + V_1L_2}_{V_1}) \div \underbrace{(V_2H \times 10^4 + V_2L)}_{V_2} \rightarrow$ R (High-place 4 digits of quotient), R + 1 (Lower-place 4 digits of quotient), R + 2 (High-place 4 digits of remainder), R + 3 (Lower-place 4 digits of remainder).	ON	OFF	OFF
		$V_2 \neq 0$ $(V_1H_1 \times 10^4 + V_1H_2) \geq V_2$	0 → R, R + 1, R + 2, R + 3	OFF	ON	OFF
		$V_2 = 0$	0 → R, R + 1, R + 2, R + 3	OFF	OFF	ON
ON	ON	$V_2 = (V_2H \times 10^4 + V_2L) \neq 0$, $(\underbrace{V_1H_1 \times 10^4 + V_1H_2}_{\text{Higher-place 8 digits of } V_1}) < V_2$	$(\underbrace{V_1H_1 \times 10^{12} + V_1H_2 \times 10^5 + V_1L_1 \times 10^4 + V_1L_2}_{V_1}) \div \underbrace{(V_2H \times 10^4 + V_2L)}_{V_2} \rightarrow$ R (High-place 4 digits of integer quotient), R + 1 (Lower-place 4 digits of integer quotient), R + 2 (High-place 4 digits of decimal quotient), R + 3 (Lower-place 4 digits of decimal quotient).	ON	OFF	OFF
		$V_2 \neq 0$ $(V_1H_1 \times 10^4 + V_1H_2) \geq V_2$	0 → R, R + 1, R + 2, R + 3	OFF	ON	OFF
		$V_2 = 0$	0 → R, R + 1, R + 2, R + 3	OFF	OFF	ON
OFF	ON, OFF	None	Not operated.	OFF	OFF	OFF

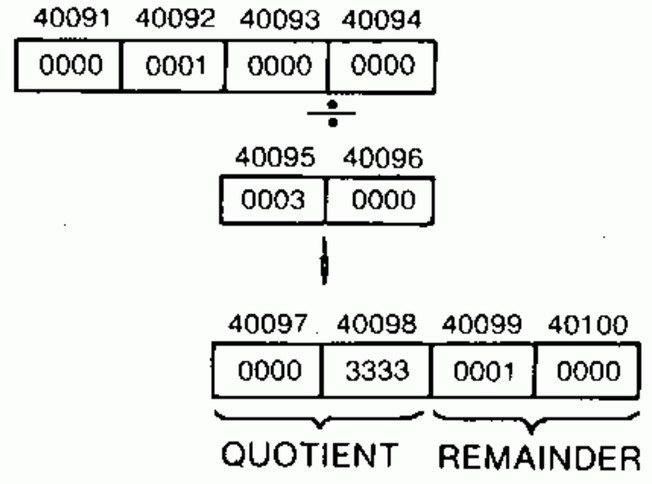
(4) Example

Example 1:



(a) Ladder

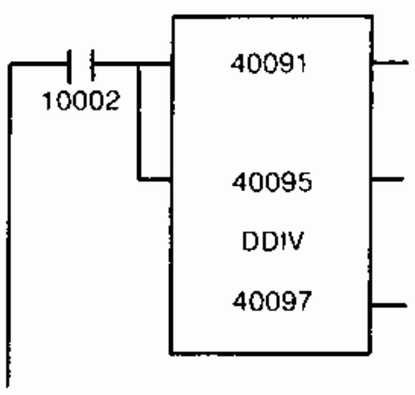




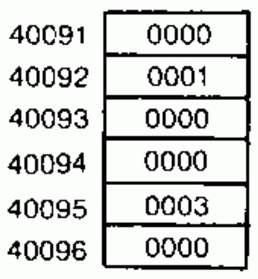
(b) DDIV Operation

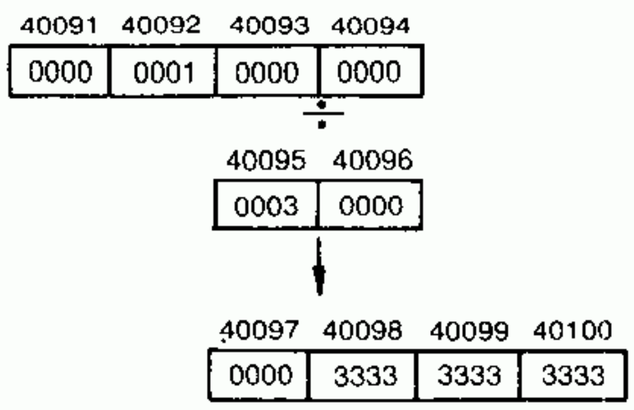
DDIV in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned ON. The result remains in 40097 to 40100 even after input relay 10001 is turned OFF.

Example 2:



(a) Ladder





(b) DDIV Contents

DDIV in (a) executes the operation of (b) when input relay 10002 is ON. Only the output 1 is turned ON. The result remains in 40097 to 40100 even after input relay 10001 is turned OFF.

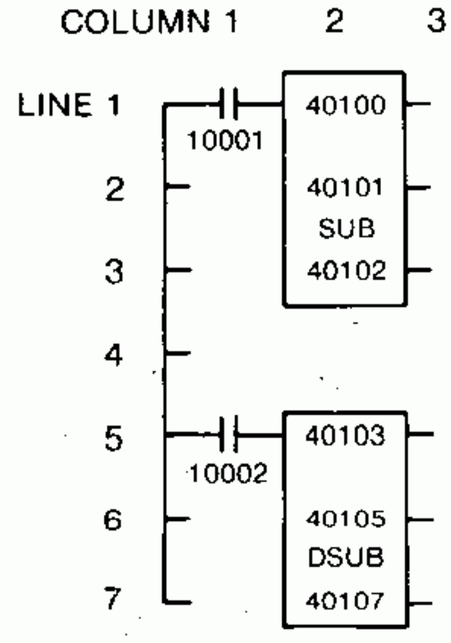
5.5.10 Programming Arithmetic Logic and Precautions

In all arithmetic operations, add, subtraction and multiply functions require only input 1, and divide requires only inputs 1 and 2. But the programming panel gives display as each output element line which may be connected to each input element line.

(1) Programming Arithmetic Logic

All arithmetic operations require three elements placed vertically (top, middle, and bottom) in a network. They can be used at any intersection of the 7 lines-by-10 columns matrix, however the top element (operand) cannot be used on either line 6 nor line 7.

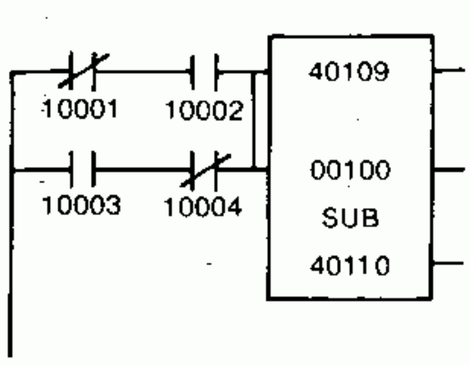
Example:



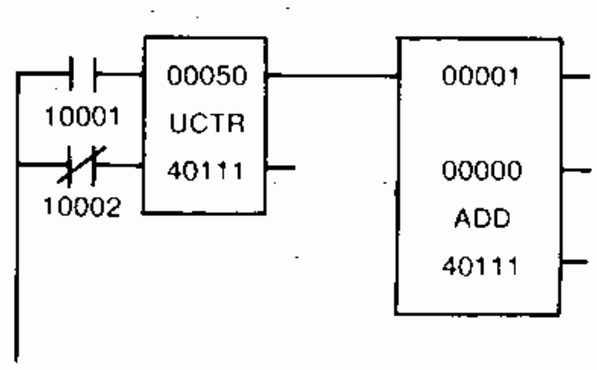
(2) Inputs of Arithmetic Logic

Inputs to the arithmetic logic may be outputs of relays, timers, counters, data processing ciruits other arithmetic operations.

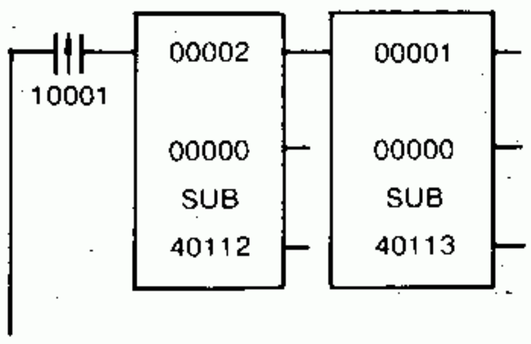
Example 1:



Example 2:



Example 3:

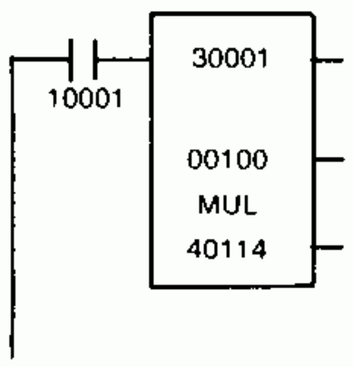


(3) Outputs of Arithmetic Logic

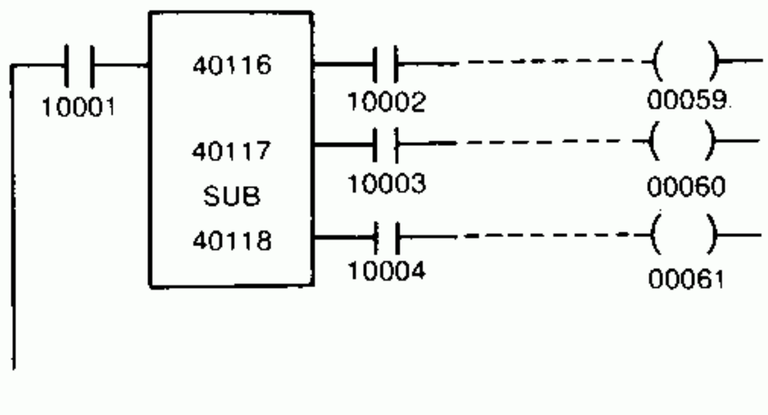
Coils need not be connected to three output nodes (1, 2, and 3) of an arithmetic function. A relay contact may be connected to the output nodes on the right or connect the output nodes directly to an input node of an arithmetic circuit, except relays.

If results of addition and double-precision addition operations exceed 9,999 or 99,999,999, respectively, only output 1 is ON. When other arithmetic circuits are cascade-connected to outputs of addition or double-precision addition, use proper care.

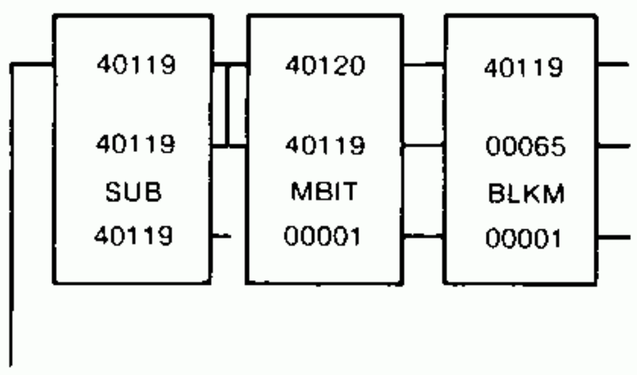
Example 1:



Example 2:



Example 3:

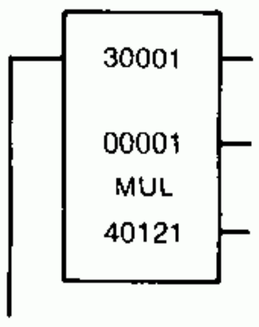


Note Subtraction operation can't be replaced with addition operation. In the case of addition, the outputs are OFF and cascade-connected arithmetic operation is not executed.

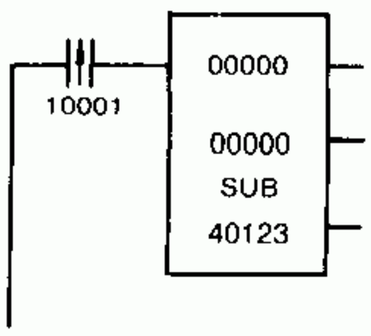
(4) Execution of Arithmetic Operations (Only One Scanning Cycle)

To execute a constant arithmetic operation, connect the inputs directly to the power rail on left. To execute it only in one scan, use a transitional contact as an input.

Example 1:



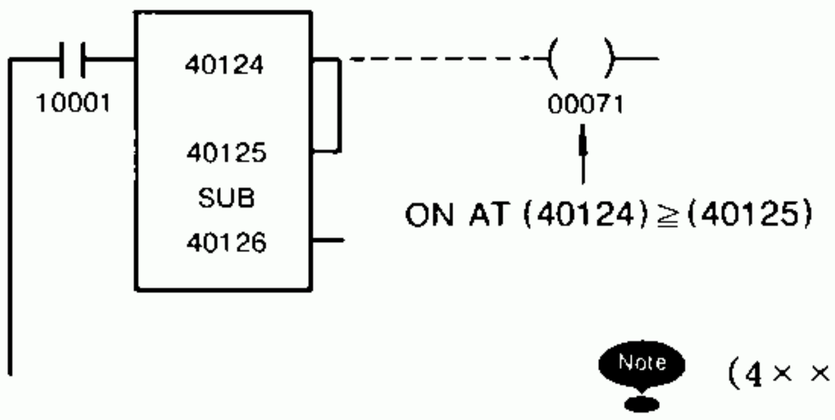
Example 2:



(5) ORed Outputs of Arithmetic Operations

Outputs of the subtraction, the double-precision subtraction, the divede, and double-precision divide can be ORed by using a vertical shunt on the output side.

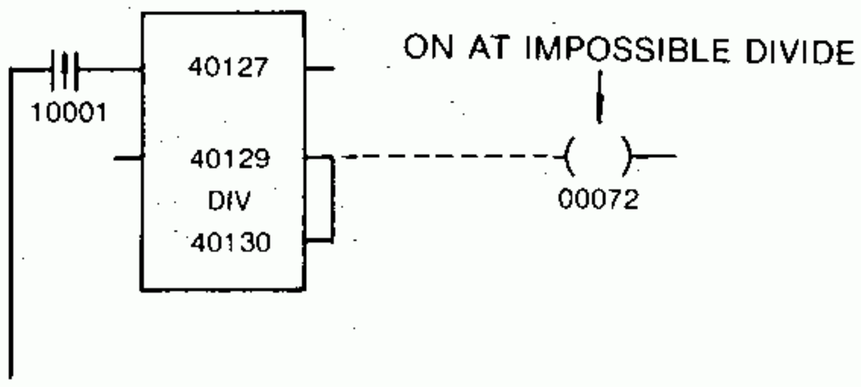
Example 1:



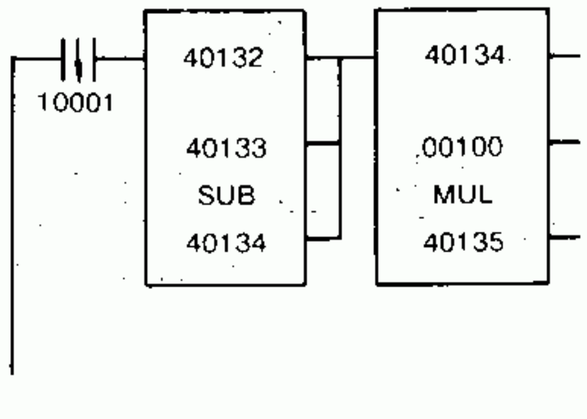
Note (4 × × × ×) ····· 4 × × × × contents

5.5.10 Programming Arithmetic Logic and Precautions (Cont'd)

Example 2:



Example 3:



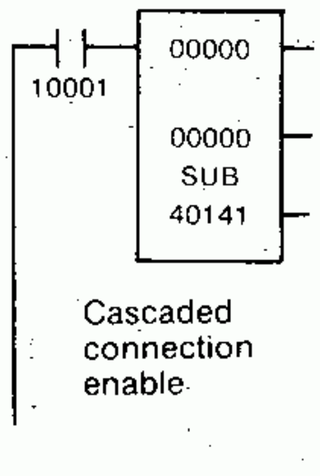
Multiply is executed regardless of the result of subtraction (cascade operation) .

5.5.11 Example-Application Circuits of Arithmetic Functions

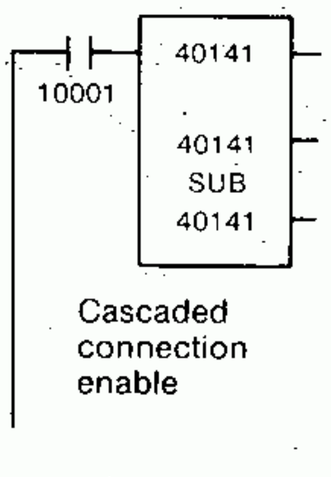
(1) Clearing Contents of Holding Registers to 0

- To clear one holding register to 0
  This is performed in three ways as shown in Examples 1 through 3 below. In all cases, the contents of 40141 become 0 when input relay 10001 is ON.

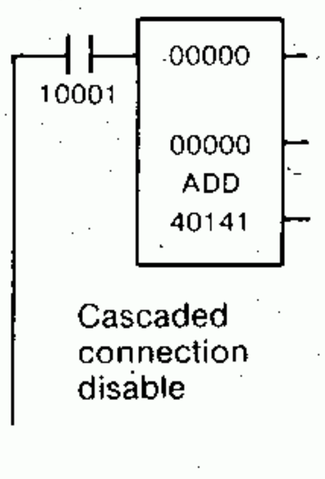
Example 1:



Example 2:

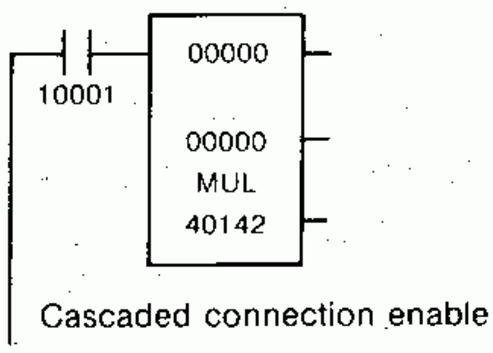


Example 3:



- To clear two successive holding registers to 0

Example:



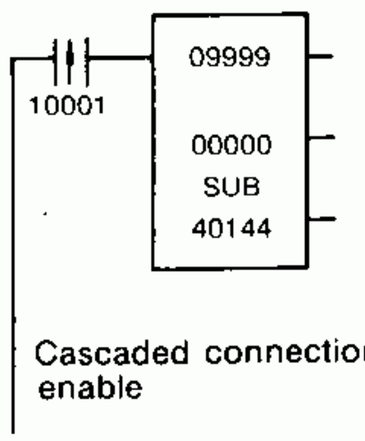
When input relay 10001 is ON, the contents of 40142 and 40143 become 0.

(2) Setting Constant or Contents of a Register to Holding Register

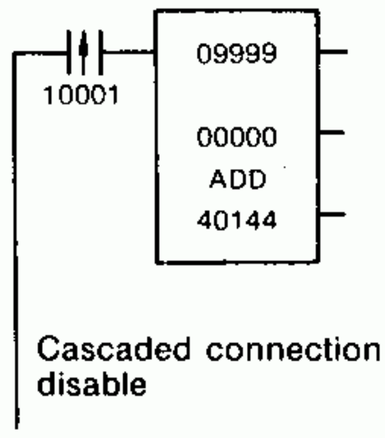
• To set the constant in a holding register

This can be performed in two ways as shown in Examples 1 and 2 below. In both cases, 9999 is set in 40144 during the scanning cycle when input relay 10001 is turned ON.

Example 1:



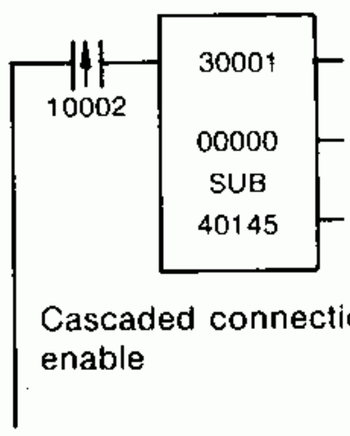
Example 2:



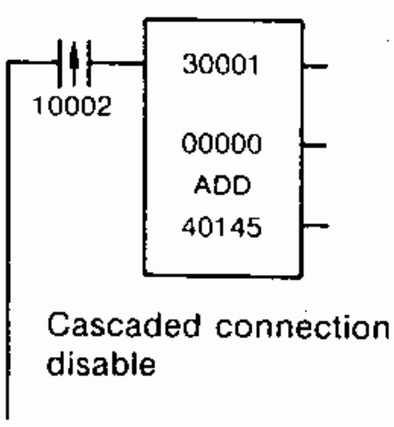
• To set the content of a register in a holding register

This can be performed in two ways as shown in Examples 1 and 2 below. In both cases, the contents of 30001 is set in 40145 during the scanning cycle when input relay 10002 is turned ON.

Example 1:



Example 2:

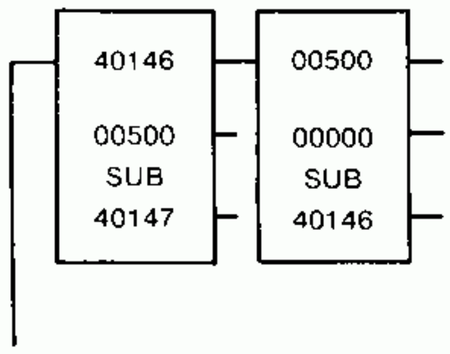


Note If the content of 30001 is greater than 10,000, it cannot be set correctly to 40145. The value of 30001 minus 10,000 is set to 40145. Use the method of Example 1 in this case.

(3) Limiting Contents of Holding Register

① To limit the maximum value

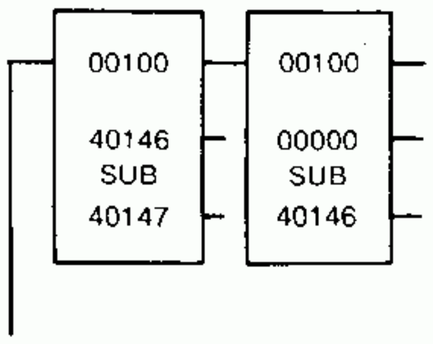
Example:



Note The maximum value of the contents of 40146 is limited to 500, i.e. (40146)≦500.

② To limit the minimum value

Example:

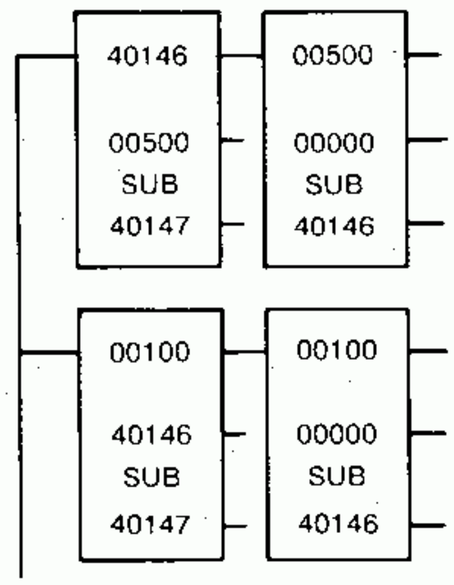


Note The minimum value of the contents of 40146 is limited to 100, i.e. 100≦(40146).

5.5.11 Example-Application Circuits of Arithmetic Functions (Cont'd)

③ To limit the range

Example:



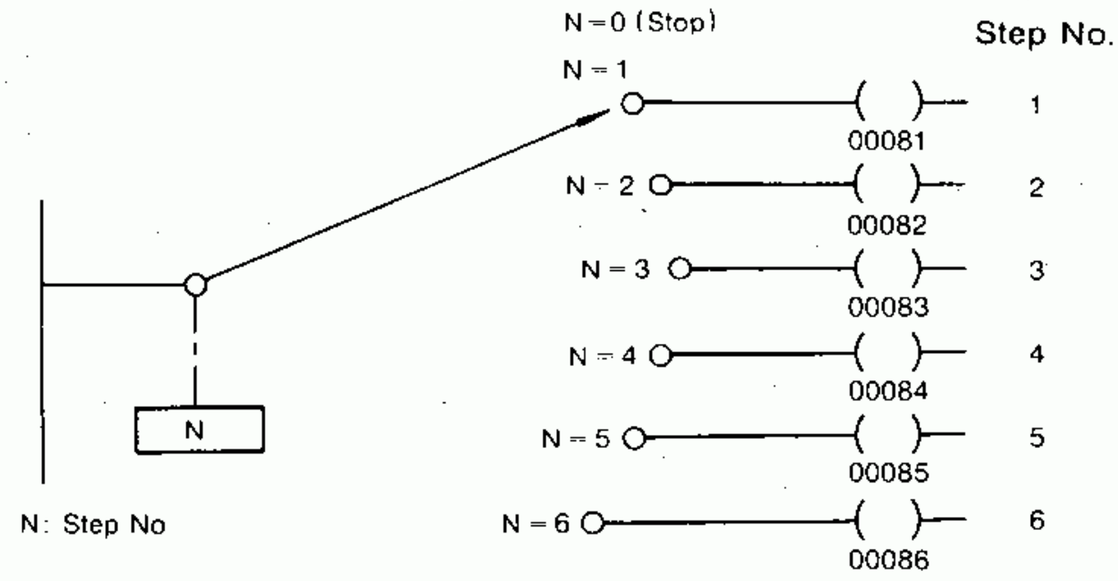
Note The range of the contents of 40146 is limited in a range of 100 to 500, i.e. $100 \leqq (40146) \leqq 500$.

(4) Stepping Switch (Having a few Steps)

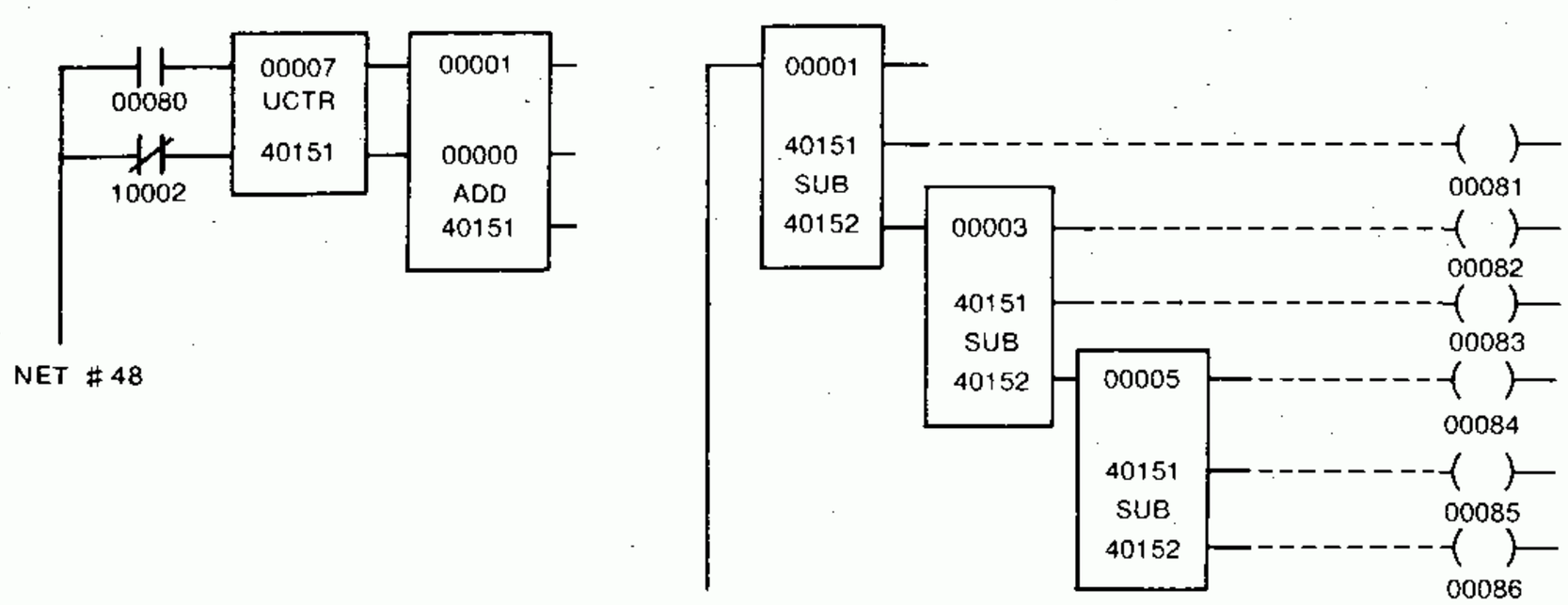
A stepping switch having a few steps can be programmed in the following way.

Example: Stepping switch with 6 steps

(a) Step Circuitry



(b) Ladder Diagram



(5) Scaling

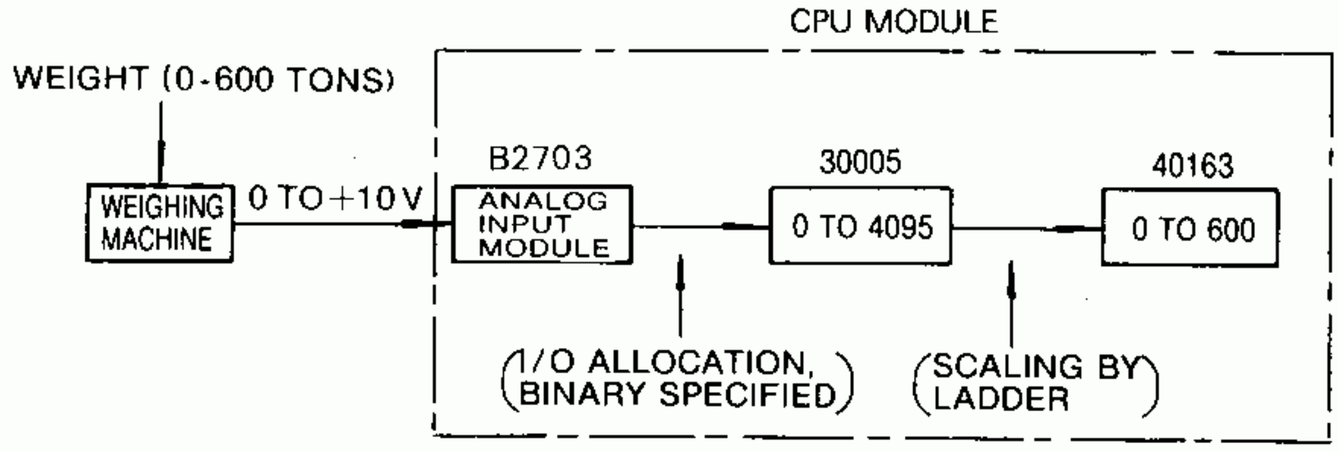
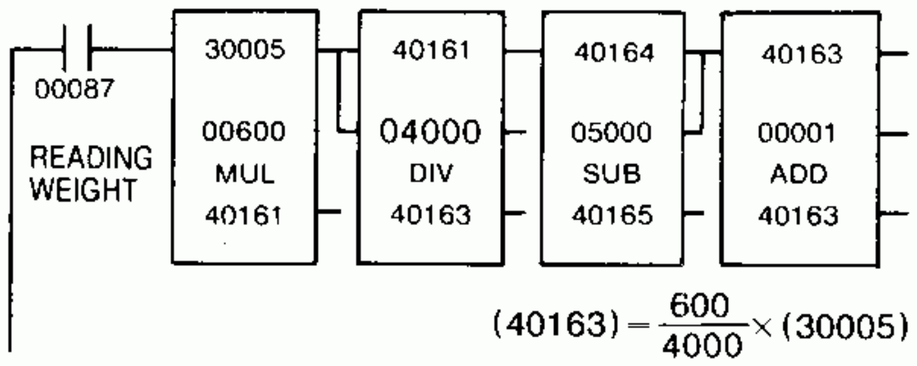


Fig. 5.37 Reading Analog Signal and Scaling

- Assume that a voltage signal of 0 to +10 V comes from the weighing machine to the analog input module B1075-1, as shown in Fig. 5.37, which converts the voltage signal to a numeric data of 0 to 4000 and enters it in input register 30005.
- The numeric data of 0 to 4000 entered in 30005 can be converted to a value of 0 to 600 (tons) by using the ladder circuit shown in Fig. 5.38.



Note The quotient of division is obtained by rounding the first decimal place.

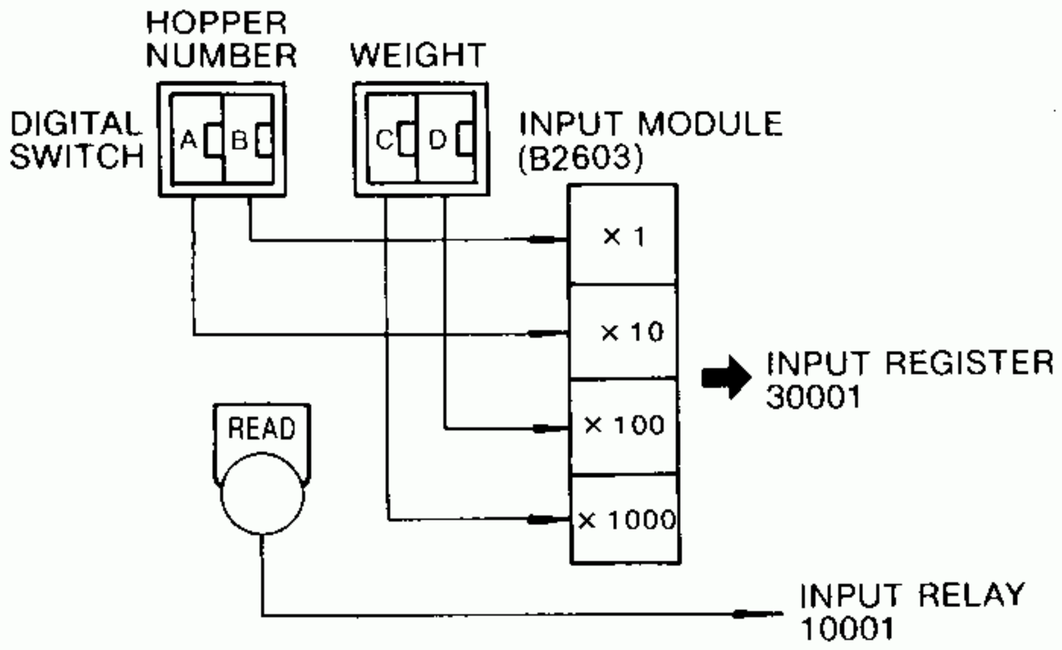
Fig. 5.38 Scaling Circuit

(6) Saving I/O Modules

It is very easy to let the GL40S read a value set by a digital switch or let a digital indicator display data stored in the GL40S by allocating the register to the I/O module. The 2000-series I/O module can deal with 4-digit BCD data. Therefore, efficiency is the highest when it deals with four BCD digits.

The following is a method to let a single module process two data having different dimensions by using multiply and divide functions. Fig. 5.39 shows an example to let an input module read two numeric data of 2 digits each. Fig. 5.40 shows an example to let an output module output numeric data of 1-and 3-digit.

① Saving input module



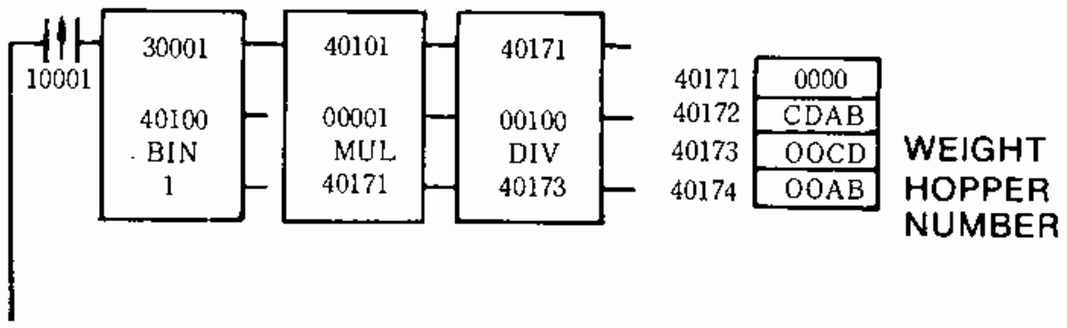


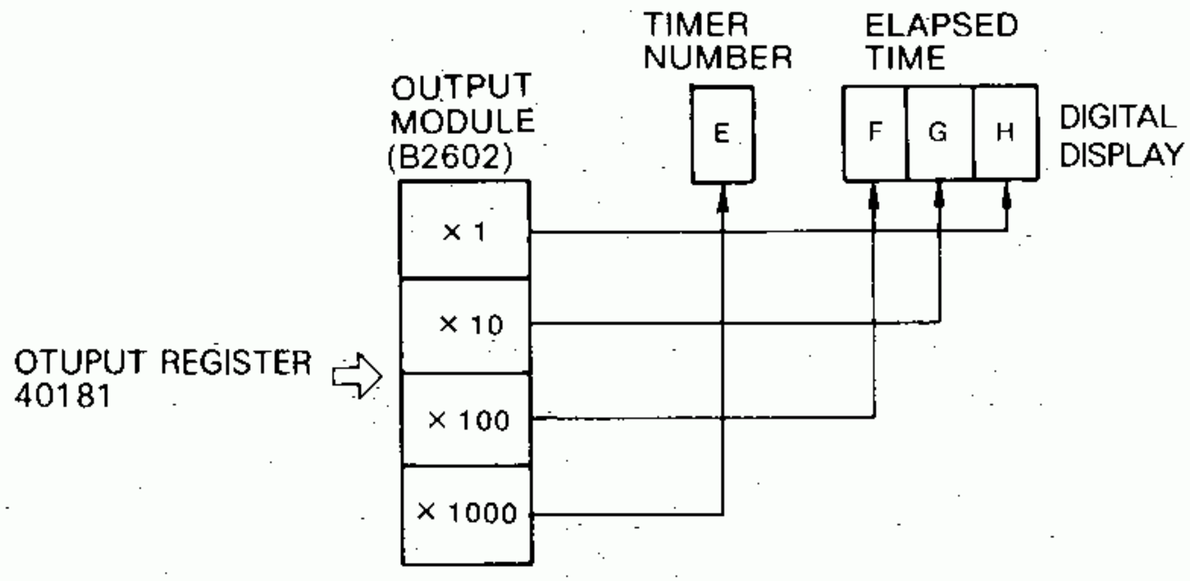
Fig. 5.39 Saving Input Module

5.5.11 Example-Application Circuits of Arithmetic Functions (Cont'd)

② Saving output module

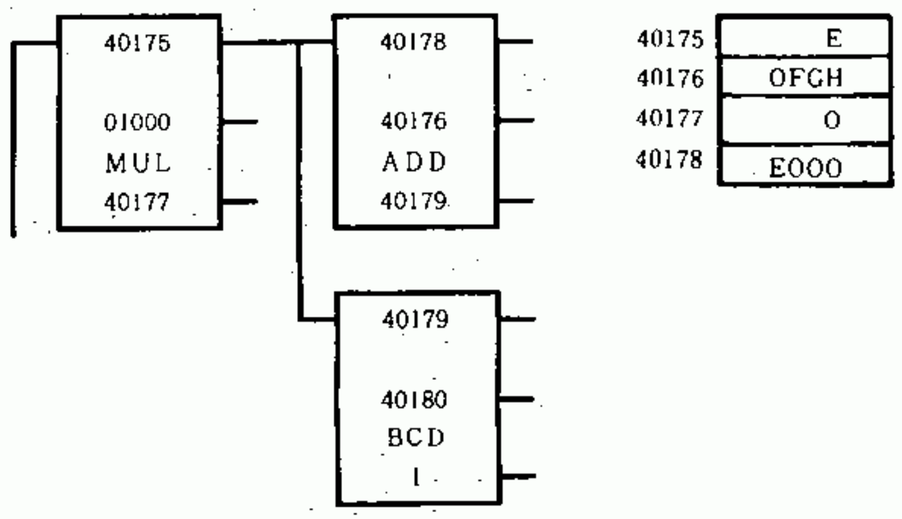
As the contents of 30001 is multiplied by the constant 1, 0 is entered in 40171 and CDAB (= C × 1000 + D × 100 + A × 10 + B) in 40172. We use the divide to produce a remainder. By CDAB ÷ 100 + CD with remainder AB, the weight and hopper number are separated in 40173 and 40174.

(a) Hardware Configuration



Note E, F, G and H are 0 to 9 and each digit is output by BCD code.

(b) Ladder Diagram



$$E \times 1000 + FGH = EFGH$$

(Where $0 \leqq E \leqq 9{,}000 \leqq FGH \leqq 999$)

Fig. 5.40 Having Output Module

5.6 SIGNED ARITHMETIC FUNCTIONS

5.6.1 Types of Signed Arithmetic Functions

The signed arithmetic operations are signed addition, subtraction, multiply, and divide applied on operand V_1 by operand V_2. There are six variations of arithmetic functions as described in Table 5.33.

Table 5.33 Types of Arithmetic Functions

Type Signed	Symbol	Operator	Range of Operand V_1	Range of Operand V_2	Reference Page
Signed Addition	SADD	$V_1 + V_2$	-9,999 to 9,999		105
Signed Double-precision Addition	SDAD		-99,999,999 to 99,999,999		108
Signed Subtraction	SSUB	$V_1 - V_2$ • V_1 compared to V_2	-9,999 to 9,999		111
Signed Double-precision Subtraction	SDSB		-99,999,999 to 99,999,999		113
Signed Multiply	SMUL	$V_1 \times V_2$	-9,999 to 9,999		116
Signed Divide	SDIV	$V_1 \div V_2$	-9,999 to 9,999	-9,999 to 9,999	118

The range of V_1 becomes as with when two successive registers are used.

5.6.2 Signed Addition (SADD)

(1) Function

Operates signed addition in 4-digit decimal.

(2) Form

- Fig. 5.41 shows the form of SADD.
- SADD is the symbol denoting the signed addition.
- SADD operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.34, specify any of reference numbers of various registers.

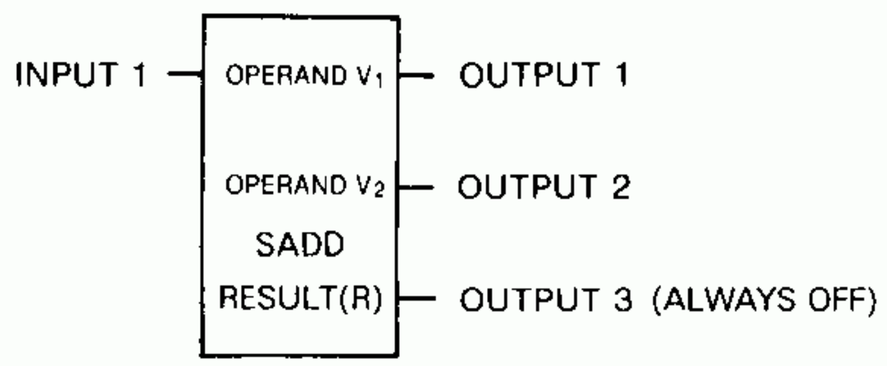


Fig. 5.41 SADD General Form

Table 5.34 Elements of SADD Function

Element Position	Specified Number	Description
Top	Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When register is specified, the contents are the operand (V_1 = -9,999 to 9,999).
Middle		• When register is specified, the contents are the operand (V_2 = -9,999 to 9,999).
Bottom	• Holding register (40001-42048) • Link register (R0001-R1024)	The result of addition function ($V_1 + V_2$ = -9,999 to 9,999) is stored in 4×××× or R××××.

(3) Operation

- By the addition (SADD), $V_1 + V_2$ is calculated when the input 1 is ON. The result is treated as follows.

  ① If $0 \leqq V_1 + V_2 \leqq 9{,}999$,

  $V_1 + V_2$ is stored in R. All outputs remain OFF.

  ② If $V_1 + V_2 \geqq 10{,}000$, $V_1 + V_2 - 10{,}000$ is stored in R. The output 1 remains OFF and the output 2 is turned ON.

  ③ If $-9{,}999 \leqq V_1 + V_2 \leqq -1$,

  $V_1 + V_2$ is stored in R. The output 1 is turned ON and output 2 remains OFF.

  ④ If $V_1 + V_2 \leqq -10{,}000$,

  $V_1 + V_2 + 10{,}000$ is stored in R.

  The outputs 1 and 2 are turned ON.

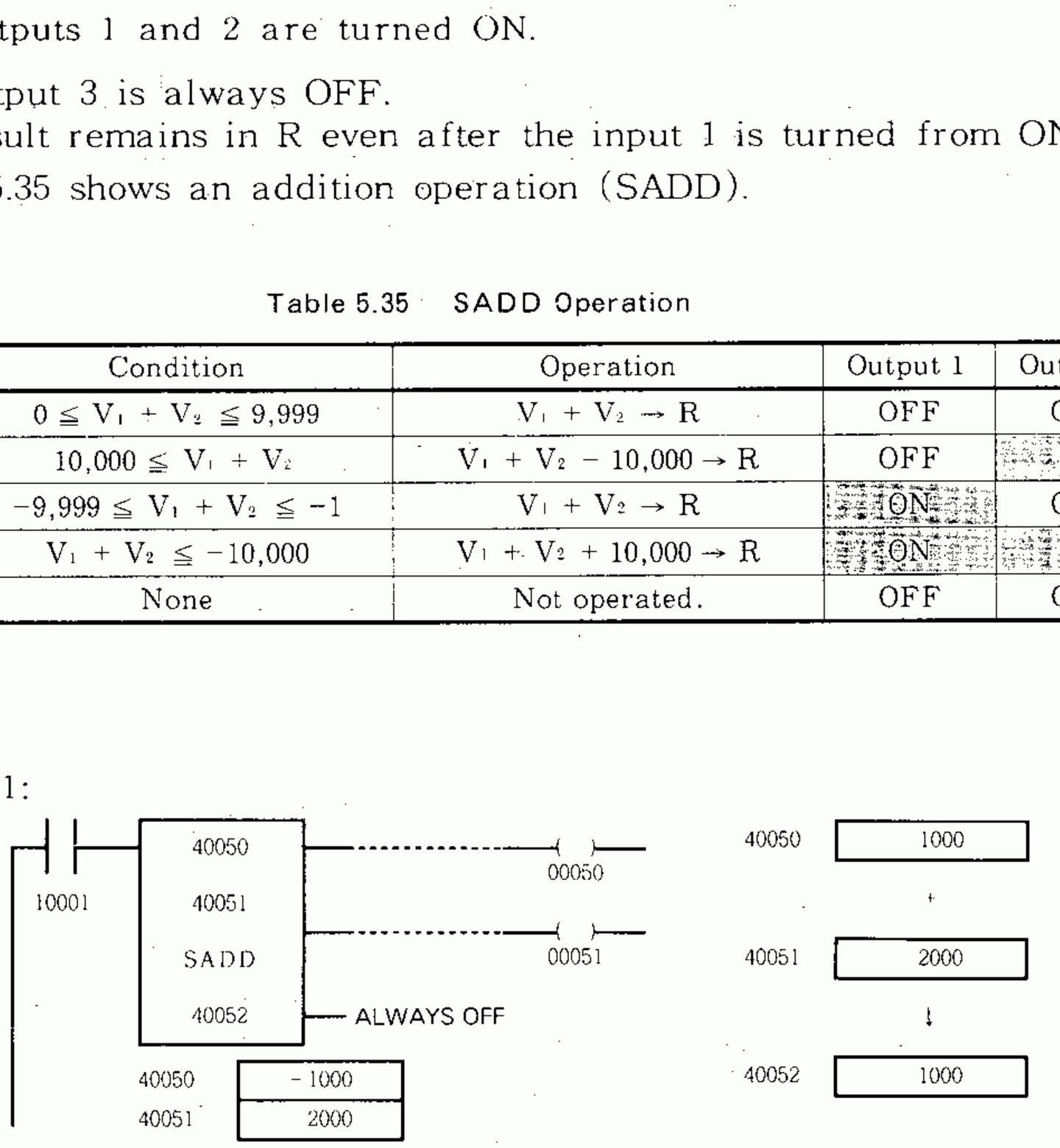
- The output 3 is always OFF.
- The result remains in R even after the input 1 is turned from ON to OFF.
- Table 5.35 shows an addition operation (SADD).

Table 5.35 SADD Operation

Input 1	Condition	Operation	Output 1	Output 2
ON	$0 \leqq V_1 + V_2 \leqq 9{,}999$	$V_1 + V_2 \rightarrow R$	OFF	OFF
	$10{,}000 \leqq V_1 + V_2$	$V_1 + V_2 - 10{,}000 \rightarrow R$	OFF	ON
	$-9{,}999 \leqq V_1 + V_2 \leqq -1$	$V_1 + V_2 \rightarrow R$	ON	OFF
	$V_1 + V_2 \leqq -10{,}000$	$V_1 + V_2 + 10{,}000 \rightarrow R$	ON	ON
OFF	None	Not operated.	OFF	OFF

(4) Example

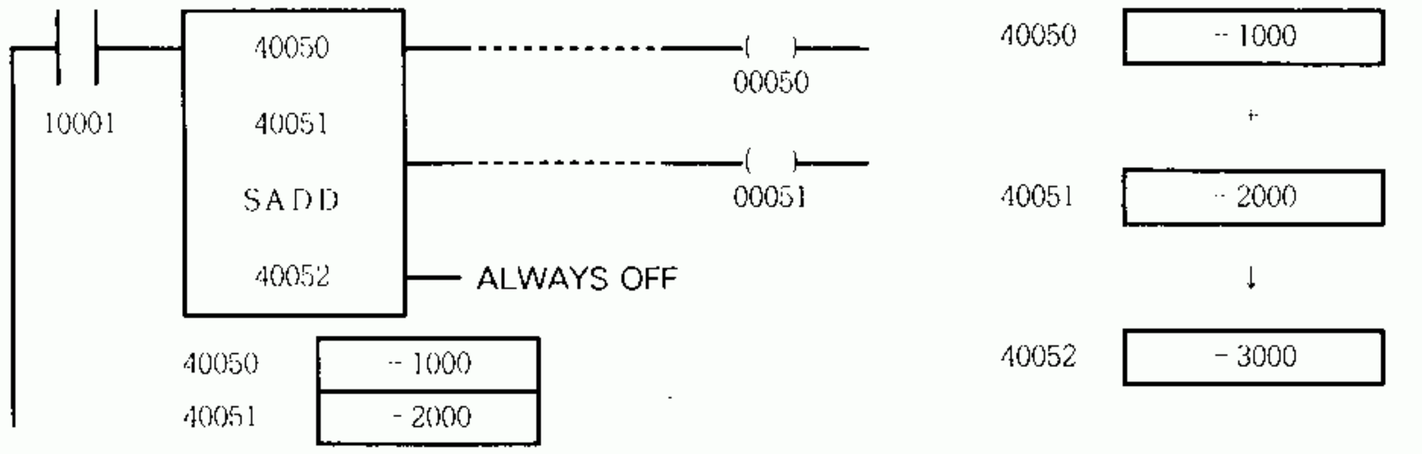
Example 1:



(a) Ladder (b) SADD Operation

SADD in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 remain OFF. The result remains in 40052 even after input relay 10001 is turned OFF.

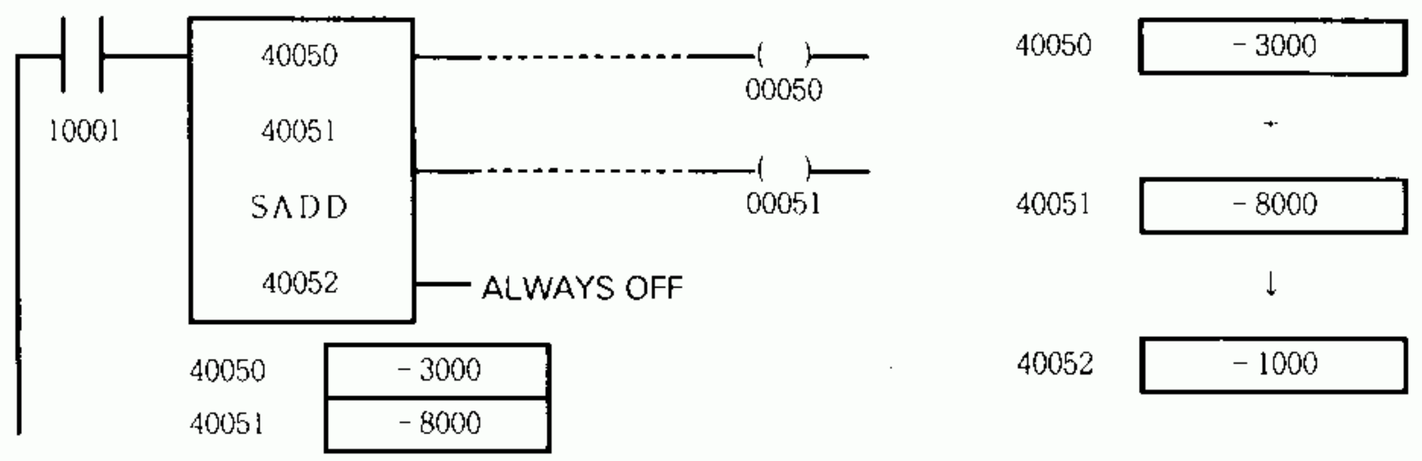
Example 2:



(a) Ladder (b) SADD Operation

SADD in (a) executes the operation of (b) is turned ON when input relay 10001 is ON. The output 1 (coil 00050) and output 2 remains OFF. Even when input relay 10001 turns from ON to OFF, the operation result remains in 40052. The data stored in 40052 are shown in the internal representation as 1000 1011 1011 1000.

Example 3:



(a) Ladder (b) SADD Operation

SADD in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 are turned ON. Thus the coils 00050 and 00051 are turned ON. The result remains in 40052 even after input relay 10001 is turned OFF.

5.6.3 Signed Double-precision Addition (SDAD)

(1) Function

Operates signed double-precision addition in 8-digit decimal.

(2) Form

- Fig. 5.42 shows the form of signed double-precision addition (SDAD).
- SDAD is the symbol denoting the signed double-precision addition.
- SDAD operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.36, specify any of reference numbers of various registers.

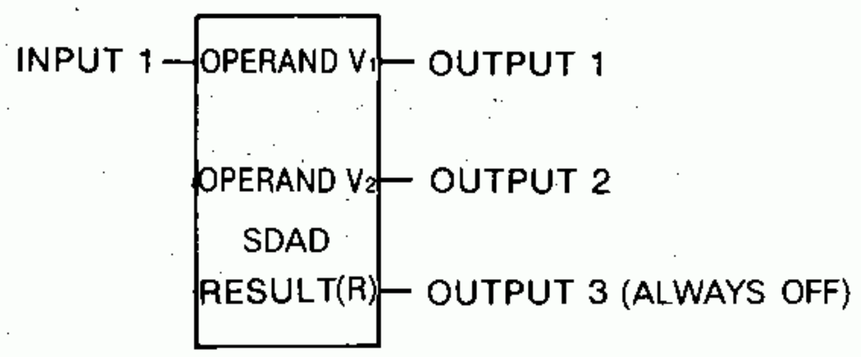


Fig. 5.42 SDAD General Form

Table 5.36 Elements of SDAD Function

Element Position	Specified Number	Description
Top	Either one of the following: • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	Operand (V_1 = −99,999,999 to 99,999,999) is stored as follows. ×××× ×××××+1 [V_1H] [V_1L] V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle		Operand (V_2 = −99,999,999 to 99,999,999) is stored as follows. ××××× ××××××+1 [V_2H] [V_2L] V_2H: Higher-place 4 digits of V_2 V_2L: Lower-place 4 digits of V_2
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of operation (V_1+V_2 = −99,999,999 to 99,999,999) is stored as follows. Example, ××××× ××××××+1 [(V_1+V_2) H] [(V_1+V_2) L] (V_1+V_2) H: Higher-place 4 digits of (V_1+V_2) (V_1+V_2) L: Lower-place 4 digits of (V_1+V_2)

(3) Operation

- By the SDAD, $V_1 + V_2$ is calculated when the input 1 is ON. The result is treated as follows.

  ① If $0 \leqq V_1 + V_2 \leqq 99,999,999$,
  $V_1 + V_2$ is stored in R and R+1. All outputs remain OFF.

  ② If $V_1 + V_2 \geqq 100,000,000$,
  $V_1 + V_2 - 1,000,000,000$ is stored in R and R+1.
  The output 1 remains OFF and the output 2 is turned ON.

  ③ If $-99,999,999 \leqq V_1 + V_2 \leqq -1$,
  $V_1 + V_2$ is stored in R and R+1.
  The output 1 is turned ON and the output 2 remains OFF.

  ④ If $V_1 + V_2 \leqq -100,000,000$,
  $V_1 + V_2 + 100,000,000$ is stored in R and R+1.
  The outputs 1 and 2 are turned ON.

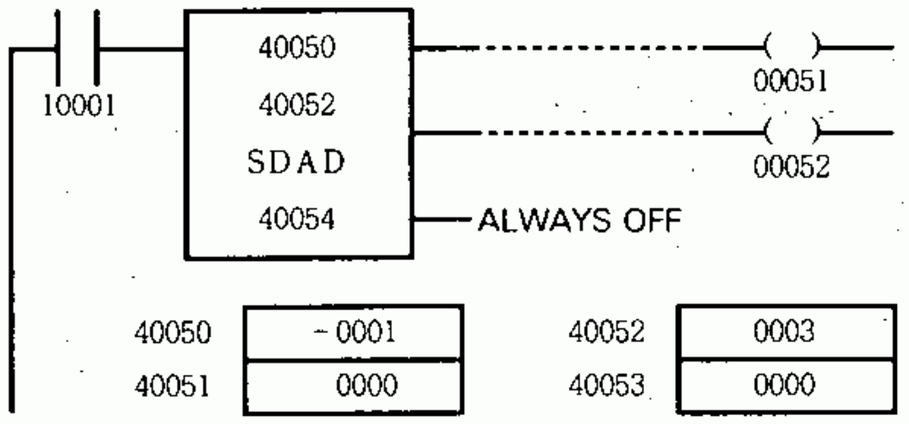
- The output 3 is always OFF.
- The result remains in R and R+1 even after the input 1 is turned from ON to OFF.
- Table 5.37 shows the SDAD.

Table 5.37 SDAD Operation

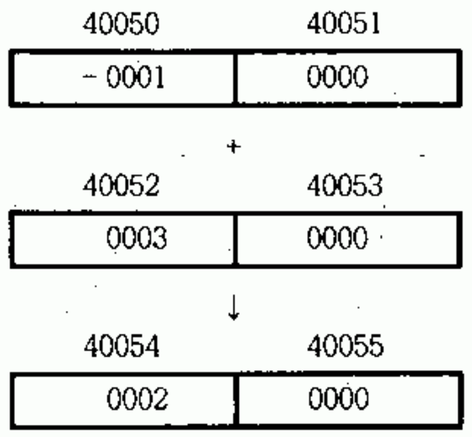
Input 1	Condition	Operation	Output 1	Output 2
ON	$0 \leqq V_1 + V_2 \leqq 99,999,999$	$V_1 + V_2 \rightarrow R$	OFF	OFF
	$100,000,000 \leqq V_1 + V_2$	$V_1 + V_2 - 100,000,000 \rightarrow R$	OFF	ON
	$-99,999,999 \leqq V_1 + V_2 \leqq -1$	$V_1 + V_2 \rightarrow R$	ON	OFF
	$V_1 + V_2 \leqq -100,000,000$	$V_1 + V_2 + 100,000,000 \rightarrow R$	ON	ON
OFF	None	Not operated.	OFF	OFF

(4) Example

Example 1:



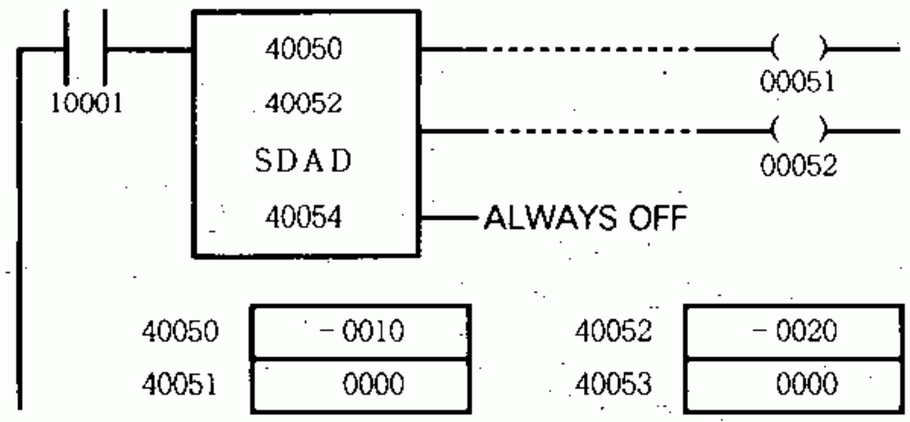
(a) Ladder



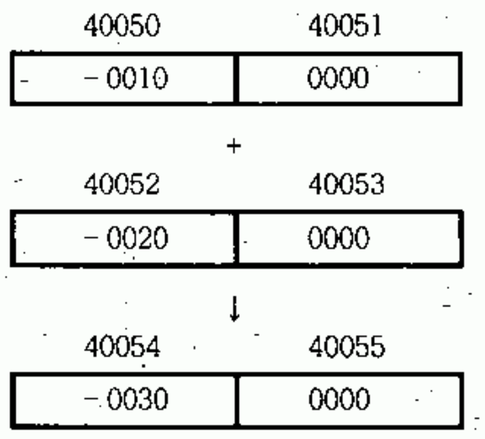
(b) SDAD Operation

SDAD in (a) exectutes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 remain OFF. The results remain in 40054 and 40055 even after input relay 10001 is turned OFF.

Example 2:



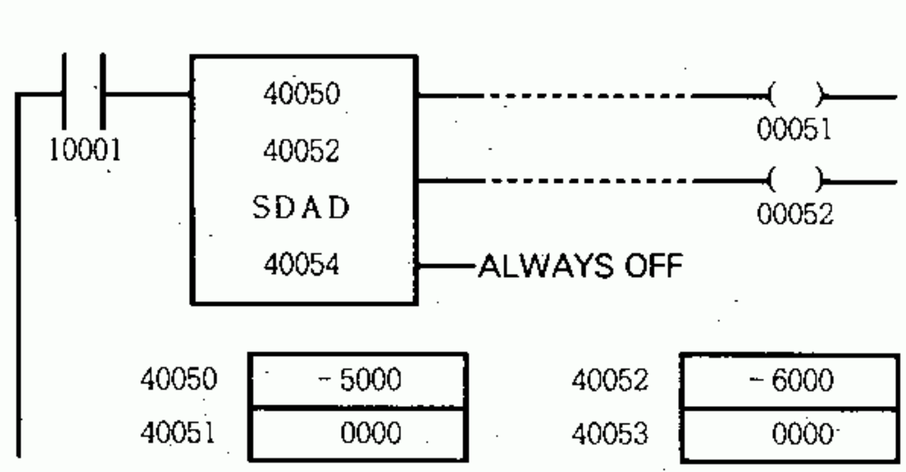
(a) Ladder



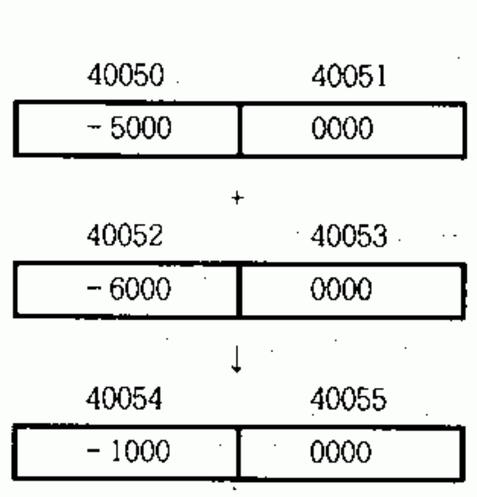
(b) SDAD Operation

SDAD in (a) executes the operation of (b) when input relay 10001 is ON. The output 2 remains OFF. The output 1 (coil 00051) is turned ON. The result remains in 40054 and 40055 even after input relay 10001 is turned OFF.

Example 3:



(a) Ladder



(b) SDAD Operation

SDAD in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 are ON. Accordingly, the coils 00051 and 00052 are ON the output 3 remains OFF. The result remains in 40054 and 40055 even after input relay 10001 is turned OFF.

5.6.4 Signed Subtraction (SSUB)

(1) Function

Operates signed subtraction in 4-digit decimal.

(2) Form

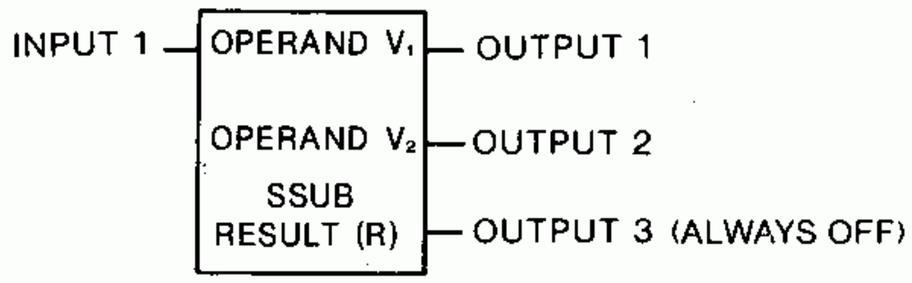


Fig. 5.43 SSUB General Form

- Fig. 5.43 shows the form of signed subtraction (SSUB).
- SSUB is the symbol denoting the signed subtraction.
- SSUB operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.38, specify any of the reference numbers of various registers.

Table 5.38 Elements of SSUB Function

Element Position	Specified Number	Description
Top	Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When the registers are specified, the contents are the operand (V_1 = -9,999 to 9,999).
Middle		• When the registers are specified, the contents are the operand (V_2 = -9,999 to 9,999).
Bottom	• Holding register (40001-42048) • Link register (R0001-R1024)	The result of SSUB function ($V_1 - V_2$ = -9,999 to to 9,999) is stored in 4×××× or R××××.

(3) Operation

- By the SSUB, $V_1 - V_2$ will be calculated when the input 1 is ON. The result is treated as follows.

① If $0 \leqq V_1 - V_2 \leqq 9{,}999$,

$V_1 - V_2$ is stored in R and all the outputs remain OFF.

② If $V_1 - V_2 \geqq 10{,}000$,

$V_1 - V_2 - 10{,}000$ is stored in R, the output 1 remains OFF and the output 2 is turned ON.

③ If $-9{,}999\ V_1 - V_2 \leqq -1$,

$V_1 - V_2$ is stored in R, the output 1 is turned ON and the output 2 remains OFF.

④ If $V_1 - V_2 \leqq -10{,}000$,

$V_1 - V_2 + 10{,}000$ is stored in R, and both of the outputs 1 and 2 are turned ON.

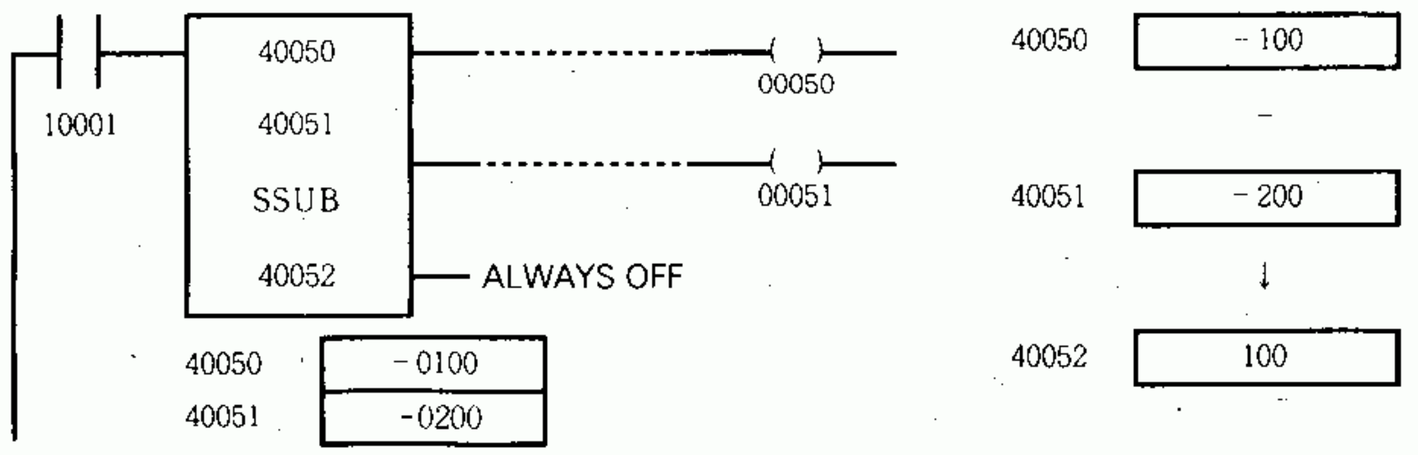
- The output 3 is always OFF.
- The result remains in even after the input 1 is turned from OFF to ON.
- Table 5.39 shows a SSUB operation.

Table 5.39 SSUB Operation

Input 1	Condition	Operation	Output 1	Output 2
ON	$0 \leqq V_1 - V_2 \leqq 9{,}999$	$V_1 - V_2 \rightarrow R$	OFF	OFF
	$10{,}000 \leqq V_1 - V_2$	$V_1 - V_2 - 10{,}000 \rightarrow R$	OFF	ON
	$-9{,}999 \leqq V_1 - V_2 \leqq -1$	$V_1 - V_2 \rightarrow R$	ON	OFF
	$V_1 - V_2 \leqq -10{,}000$	$V_1 - V_2 + 10{,}000 \rightarrow R$	ON	ON
OFF	None	Not operated.	OFF	OFF

(4) Example

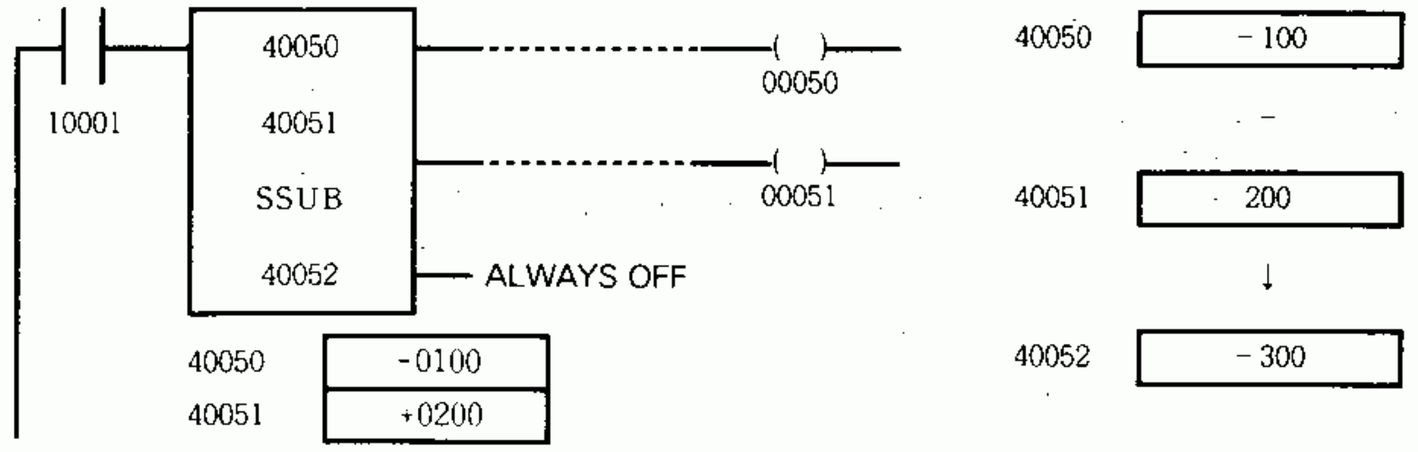
Example 1:



(a) Ladder (b) SSUB Operation

SSUB in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 remain OFF. The result remains in 40052 even after input relay 10001 is turned OFF.

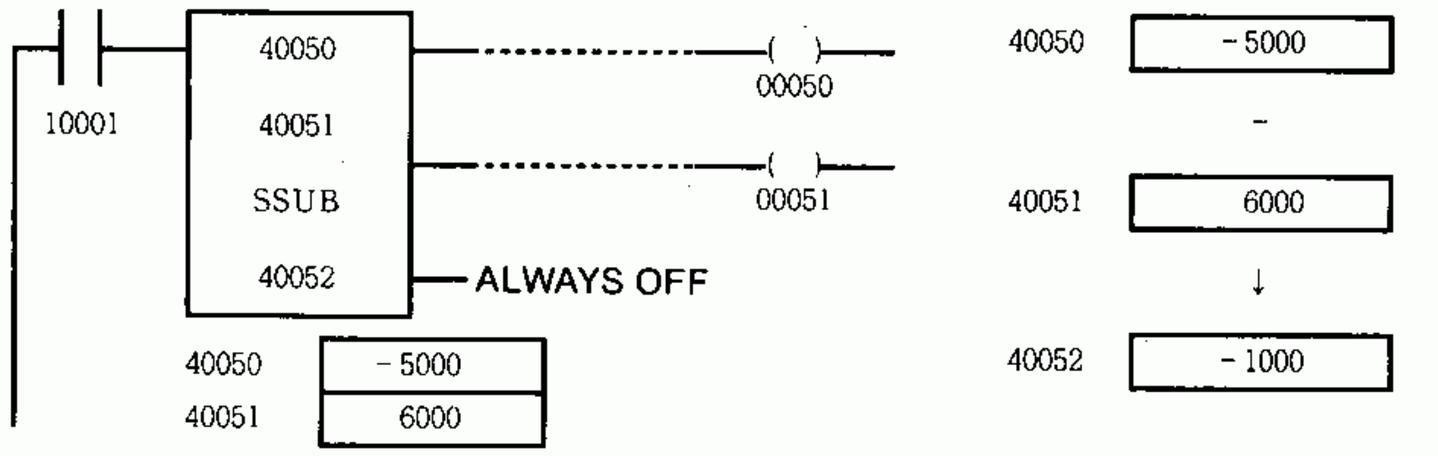
Example 2:



(a) Ladder (b) SSUB Operation

SSUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 (coil 00050) is turned ON. The output 2 remains OFF. The result remains in 40052 even after input relay 10001 is turned OFF.

Example 3:



(a) Ladder (b) SSUB Operation

SSUB in (a) executes the operation of (b) when input relay 10001 is ON. Only the outputs 1 and 2 (coils 00050 and 00051) is turned ON. The result remains in 40052 even after input relay 10001 is turned OFF.

5.6.5 Signed Double-precision Subtraction (SDSB)

(1) Function

Operates signed double-precision subtraction in 8-digit decimal.

(2) Form

- Fig. 5.44 shows the form of signed double-precision subtraction (SDSB).

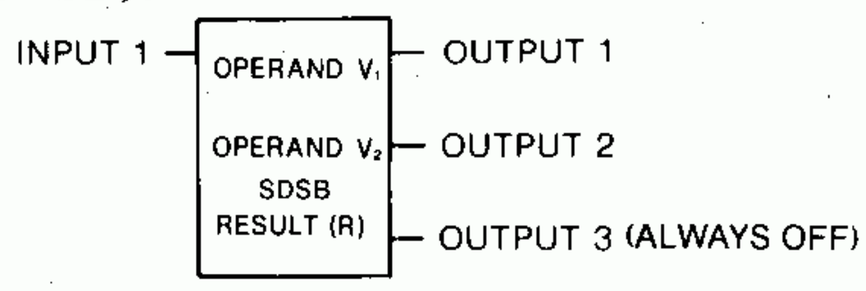


Fig. 5.44 SDSB General Form

- SDSB is the symbol denoting the signed double-precision subtraction.
- SDSB operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.40, specify either reference number for each of the top, middle and bottom elements.

Table 5.40 Elements SDSB Function

Element Position	Specified Number	Description
Top	Either one of the following: • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	Operand (V_1 = -99,999,999 to 99,999,999) is stored as follows. ×××××: V_1H ; ×××××+1: V_1L V_1H: Higher-place 4 digits of V_1. V_1L: Lower-place 4 digits of V_1.
Middle		Operand (V_2 = -99,999,999 to 99,999,999) is stored as follows. ×××××: V_2H ; ×××××+1: V_2L V_2H: Higher-place 4 digits of V_2. V_2L: Lower-place 4 digits of V_2.
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of operation (-99,999,999 to 99,999,999) is stored as follows. ×××××: $(V_1-V_2)H$; ×××××+1: $(V_1-V_2)L$ $(V_1-V_2)H$: Higher-place 4 digits of $(V_1-V_2)H$, $(V_1-V_2)L$: Lower-place 4 digits of $(V_1-V_2)L$.

(3) Operation

- By the SDSB, V_1-V_2 will be calculated when the input 1 is ON. The result is treated as input 1 is ON. The result is

  ① If $0 \leqq V_1 - V_2 \leqq 99,999,999$,

  V_1-V_2 is stored in R, and all outputs remain OFF.

  ② If $V_1 - V_2 \geqq 100,000,000$,

  V_1-V_2 − 100,000,000 is stored in R. The output 1 remains OFF and the output 2 is turned on.

  ③ If $-99,999,999 \leqq V_1 - V_2 \leqq -1$,

  $V - V_2$ is stored in R. The output 1 is turned on and the output 2 remains OFF.

  ④ If $V_1 - V_2 \leqq -100,000,000$,

  $V_1 - V_2 + 100,000,000$ is stored in R and the outputs 1 and 2 are turned on.

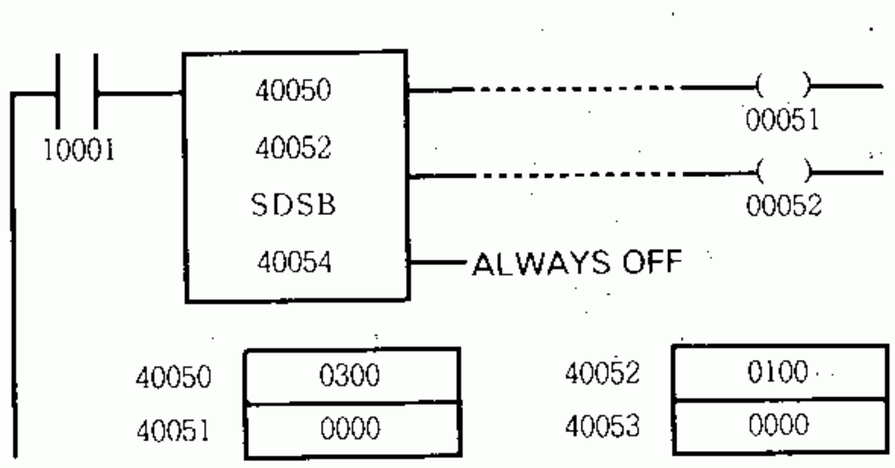
- The output 3 is always OFF.
- The result remains in R even after the input 1 is turned OFF.
- Table 5.41 shows SDSB operation.

Table 5.41 SDSB Operation

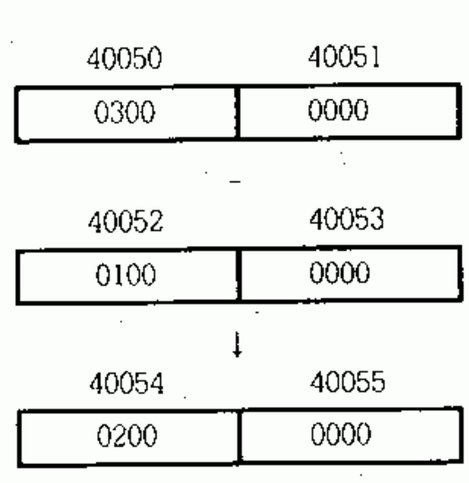
Input 1	Condition	Operation	Output 1	Output 2
ON	$0 \leqq V_1 - V_2 \leqq 99,999,999$	$V_1 - V_2 \rightarrow R$	OFF	OFF
	$100,000,000 \leqq V_1 - V_2$	$V_1 - V_2 - 100,000,000 \rightarrow R$	OFF	ON
	$-99,999,999 \leqq V_1 - V_2 \leqq -1$	$V_1 - V_2 \rightarrow R$	ON	OFF
	$V_1 - V_2 \leqq -100,000,000$	$V_1 - V_2 + 100,000,000 \rightarrow R$	ON	ON
OFF	—	Not operated.	OFF	OFF

(4) Example

Example 1:



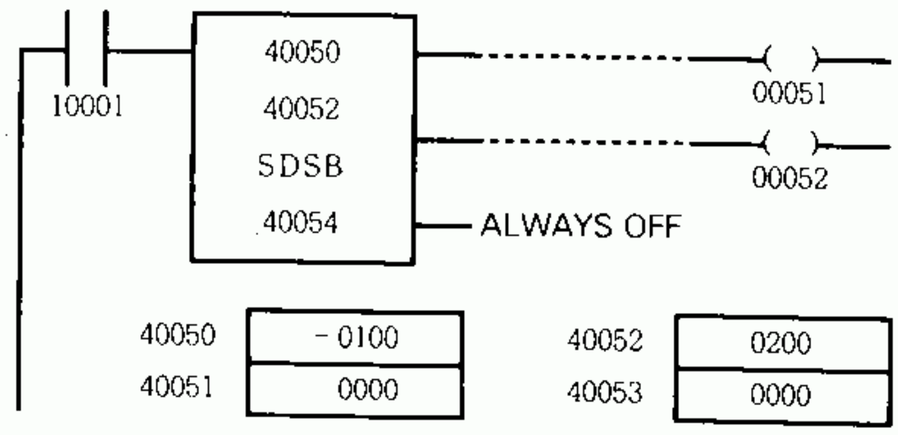
(a) Ladder



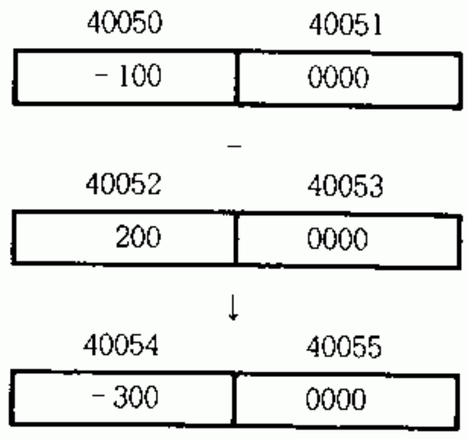
(b) SDSB Operation

SDSB in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 remain OFF. The result remains in 40054 and 40055 even after input relay 10001 is turned off.

Example 2:



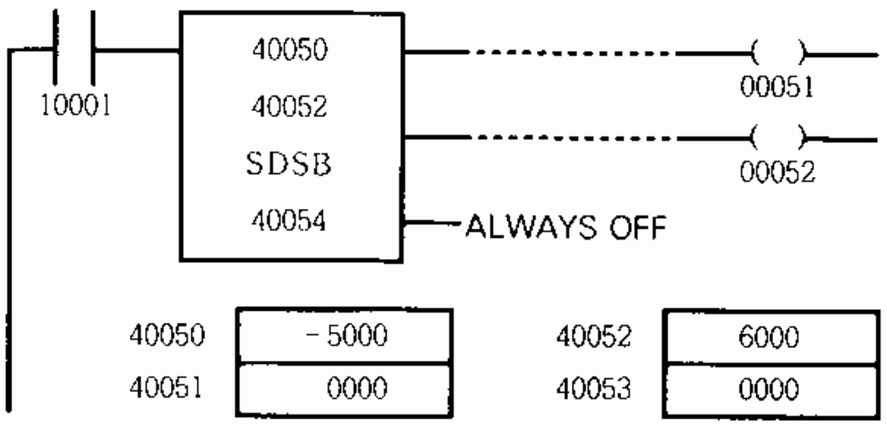
(a) Ladder



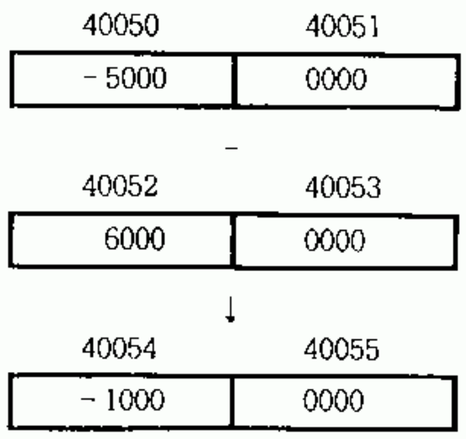
(b) SDSB Operation

SDSB in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 (coil 00051) is turned on. The output 2 remains OFF. The result remains in 40054 and 40055 even after input relay 10001 is turned off.

Example 3:



(a) Ladder



(b) SDSB Operation

SDSB in (a) executes the operation of (b) when input relay 10001 is ON. The outputs 1 and 2 (coils 00051 and 00052) are turned on. The output 3 remains OFF. The result remains in 40054 and 40055 even after input relay 10001 is turned off.

5.6.6 Signed Multiply (SMUL)

(1) Function

Operates signed multiply is 4-digit decimal.

(2) Form

- Fig. 5.45 shows the form of signed multiply (SMUL).

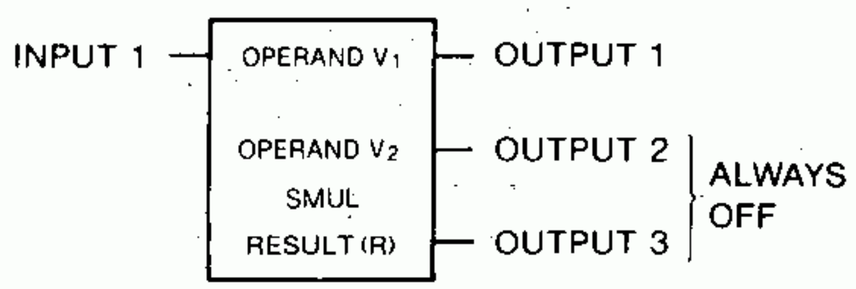


Fig. 5.45 SMUL General Form

- Multiply operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.42, specify any of reference number for each of the top, middle and bottom elements.
- SMUL is the symbol denoting the signed multiply.

Table 5.42 Elements of SMUL Function

Element Position	Specified Number	Description
Top	Any one of the following: • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When registers are specified, the contents are the operand (V_1 = 0 to 9,999).
Middle		• When registers are specified, the contents are the operand (V_2 = 0 to 9,999).
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of operation ($V_1 \times V_2$ = -99,999,999 to 99,999,999) is stored as follows : ××××× ××××× + 1 [($V_1 \times V_2$) H] [($V_1 \times V_2$) L] ($V_1 \times V_2$) H : Higher-place 4 digits of ($V_1 \times V_2$) ($V_1 \times V_2$) L : Lower-place 4 digits of ($V_1 \times V_2$)

(3) Operation

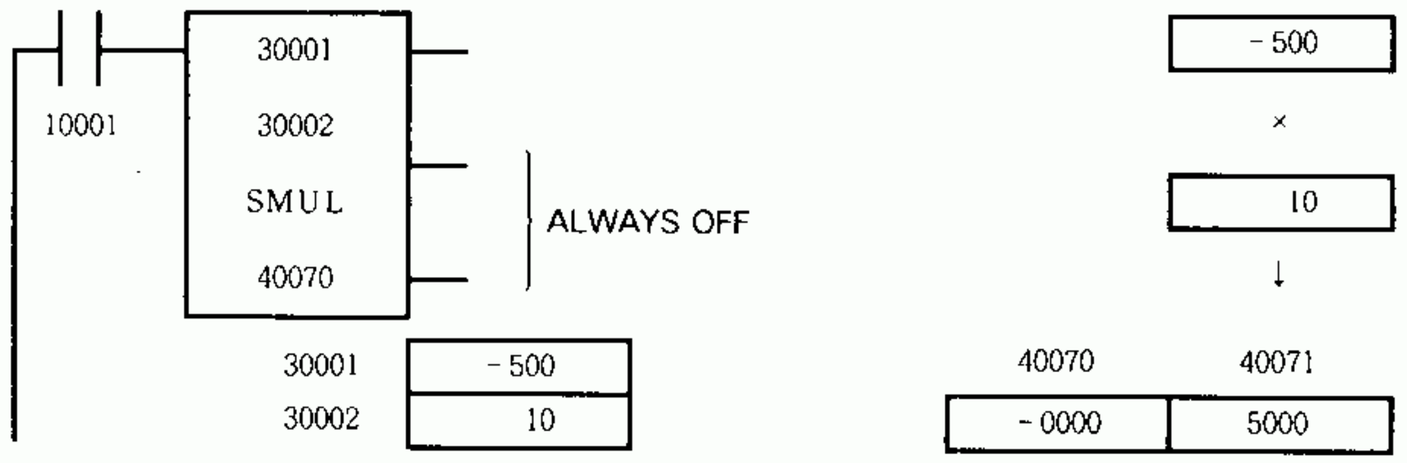
- By the SMUL, $V_1 \times V_2$ is calculated when the input 1 is ON. The result is treated as follows. The four higher-place digits of $V_1 \times V_2$ are stored in R and the four lower-place digits in R + 1. When the result is negative, the output 1 is ON. When the result is positive, the output remains OFF.
- The outputs 2 and 3 are always ON.
- Table 5.43 shows SMUL operation.

Table 5.43 SMUL Operation

Input 1	Condition	Operation	Result	Output 1
ON	None	$V_1 \times V_2 \rightarrow R$ (Higher-place 4 digits)	Positive	OFF
		$R + 1$ (Lower-place 4 digits)	Negative	ON
OFF	None	Not provided.	—	OFF

(4) Examples

Example 1:



(a) Ladder (b) SMUL Operation

SMUL in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 is turned on. The result remains in 40070 and 40071 even after input relay 10001 is turned off.

5.6.7 Signed Divide (SDIV)

(1) Function

Operates signed divide in 8-digit decimal.

(2) Form

- Fig. 5.46 shows the form of signed divide (SDIV).

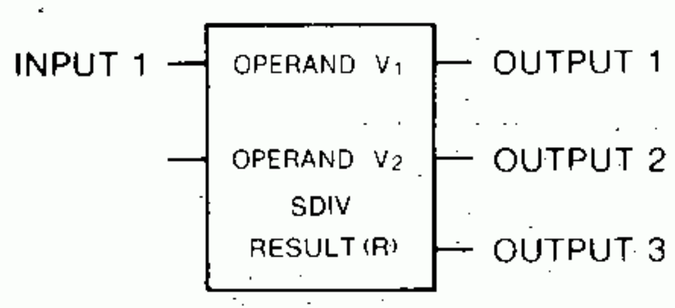


Fig. 5.46 SDIV General Form

- SDIV is the symbol denoting the signed divide.
- SDIV operation requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.44, specify any of reference numbers for each of the top, middle and bottom elements.

Table 5.44 Elements of SDIV Function

Element Position	Specified Number	Description
Top	Any one of the following : • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	• The operand (V_1 = -99,989,999 to 99,989,999) is stored as follows. ×××× / ××××+1 V_1H / V_1L V_1H: Higher-place 4 digits of V_1 V_1L: Lower-place 4 digits of V_1
Middle	Any one of the following : • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	• When registers are specified, the contents are the operand (V_2 = 1 to 9,999).
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	Result of divide function ($V_1 \div V_2$) is stored as follows. Where input 2 is OFF: ××××× (The quotient), ××××+1 (The remainder) Where input 2 is ON: ××××× (The integer quotient), ××××+1 (The decimal quotient)

(3) Operation

By the SDIV, $V_1 \div V_2$ is calculated when the input 1 is ON. The result is treated as follows.

(a) If the input 2 is OFF,

The quotient of $V_1 \div V_2$ is stored in R and the remainder in R + 1. When it is negative, the output 1 is turned on. When it is positive, all outputs are OFF.

(b) If the input 2 is ON,

The integer part of the quotient of $V_1 \div V_2$ is stored in R and the decimal part (rounded off to the fifth decimal place) in R + 1. When it is negative, the output 1 is tuned on. When it is positive, all outputs are OFF.

(c) The result remains in R and R + 1 even after the input 1 is turned from ON to OFF.

(d) In the following cases, divide operation is not executed and zero is placed in each of R and R + 1.

- When $V_2 = 0$, the output 3 is turned on.
- If the quotient or the integer part of quotient overflows in R, the output 2 is turned on.

(Example) When V_1 = 500,000 and V_2 = 10, the quotient is 50,000 which overflows in R.

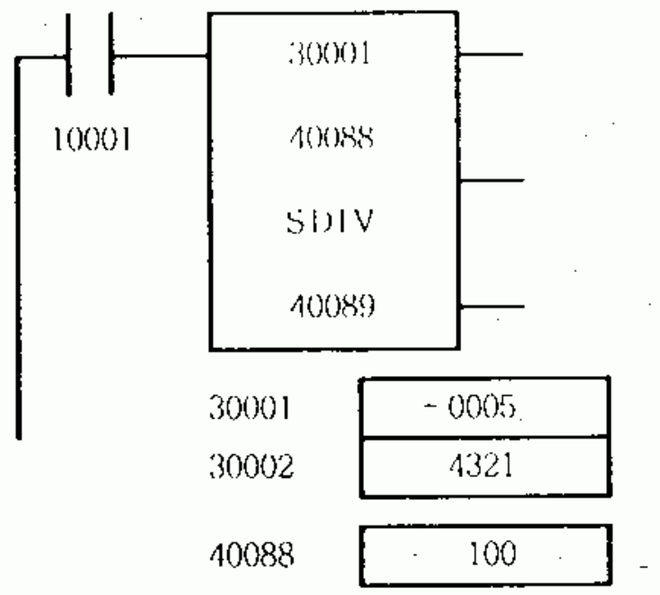
Table 5.45 shows SDIV operation.

Table 5.45 SDIV Operation

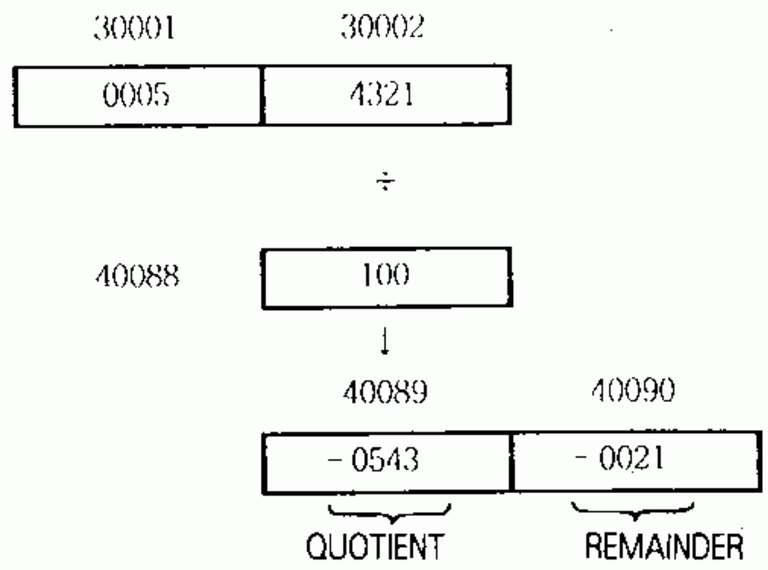
Input 1	Input 2	Condition	Operation	Result	Output 1	Output 2	Output 3
ON	OFF	$V_2 \neq 0$, $V_1H < V_2$	$(V_1H \times 10{,}000 + V_1) \div V_2$ Quotient → R Remainder R + 1	Positive	OFF	OFF	OFF
				Negative	ON	OFF	OFF
		$V_2 \neq 0$, $V_1H \geqq V_2$	0 → R R + 1		OFF	ON	OFF
		$V_2 = 0$	0 → R R + 1		OFF	OFF	ON
ON	ON	$V_2 \neq 0$, $V_1H < V_2$	$(V_1H \times 10{,}000 + V_1) \div V_2$ Integer quotient → R Decimal quotient R + 1	Positive	OFF	OFF	OFF
				Negative	ON	OFF	OFF
		$V_2 \neq 0$, $V_1H \geqq V_2$	0 → R R + 1		OFF	ON	OFF
		$V_2 = 0$	0 → R R + 1		OFF	OFF	ON
OFF	ON OFF	None	Not operated.		OFF	OFF	OFF

(4) Example

Example 1:



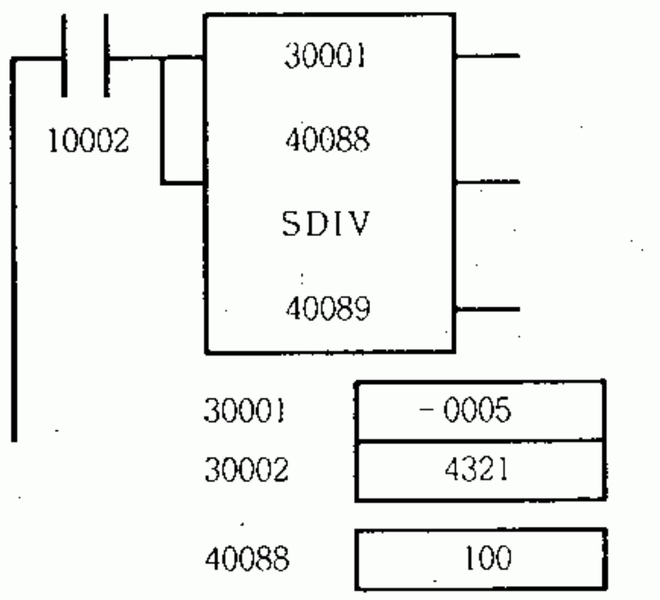
(a) Ladder



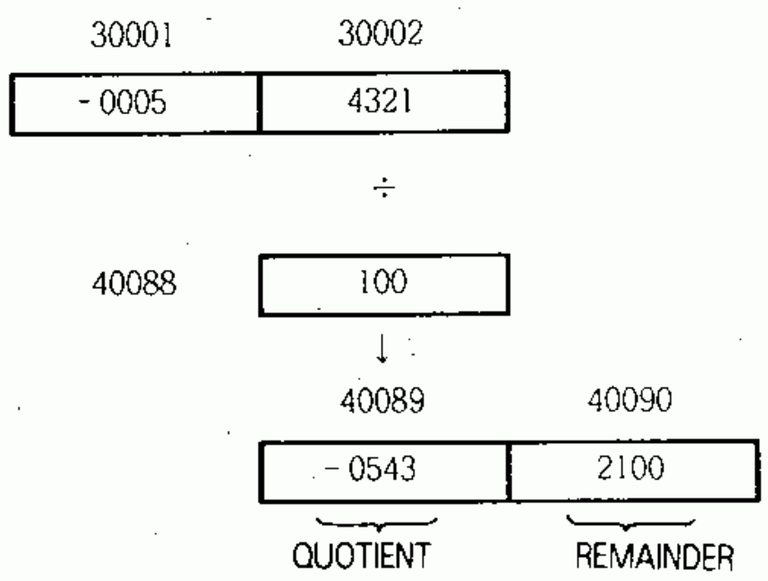
(b) SDIV Operation

SDIV in (a) executes the operation of (b) when input relay 10001 is ON. Only the output 1 is turned on. The result remains in 40089 and 40090 even after input relay 10001 is turned off.

Example 2:



(a) Ladder



(b) SDIV Operation

SDIV in (a) executes the operation of (b) when input relay 10002 is ON. Only the output 1 is turned on. The result remains in 40089 and 40090 even after input relay 10002 is turned off.

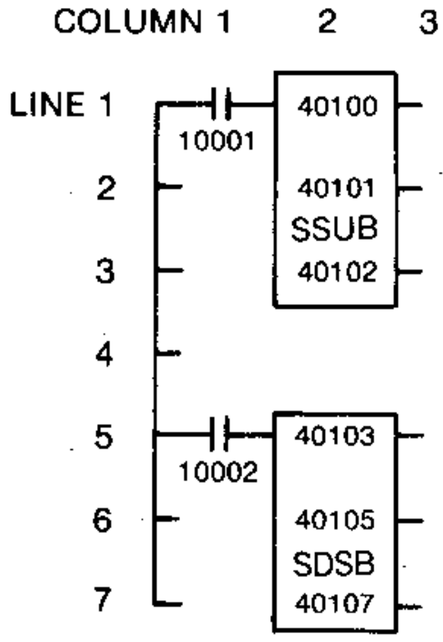
5.6.8 Programming Signed Arithmetic Logic and Precautions

In all signed arithmetic operations, add, subtraction and multiply function require only input 1, and divide requires only inputs 1 and 2. But the programming panel gives display as each output element line which may be connected to each input element line.

(1) Programming Arithmetic Logic

All signed arithmetic operations require three elements placed vertically (top, middle, and bottom) in a network. They can be used at any intersection of the 7 lines-by-10 columns matrix, however the top element (operand) cannot be used on either line 6 nor line 7.

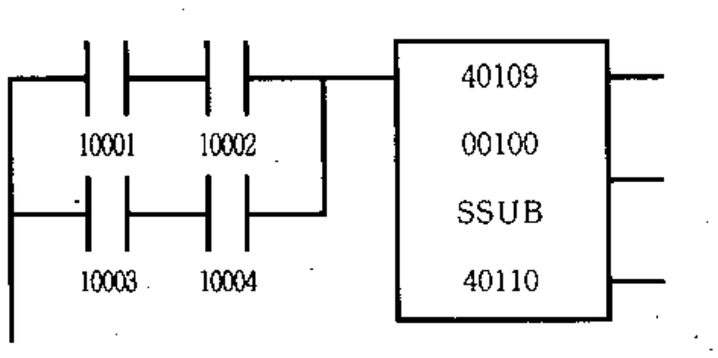
Example:



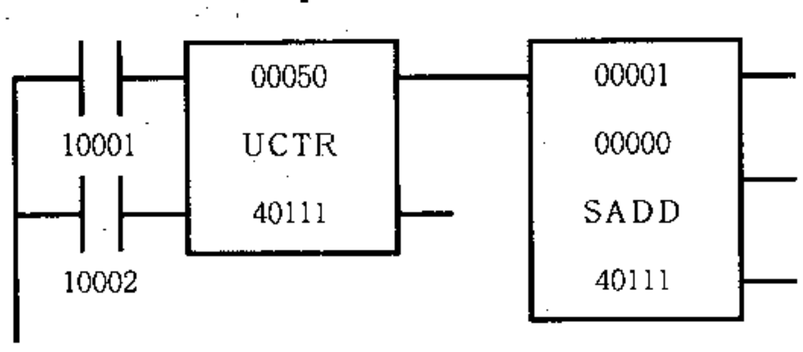
(2) Inputs of Signed Arithmetic Logic

Inputs to the signed arithmetic logic may be outputs of relays, timers, counters, data processing circuits and other arithmetic operations.

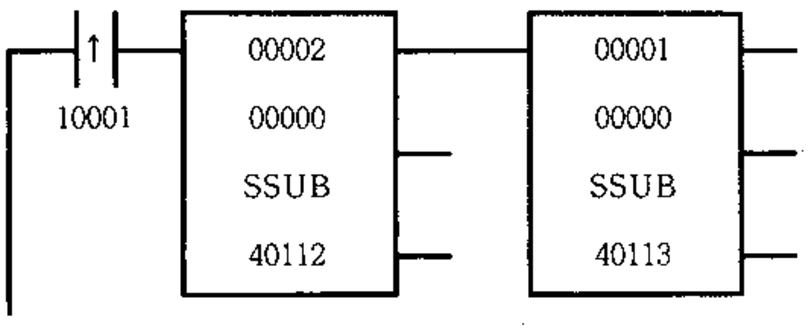
Example 1:



Example 2:



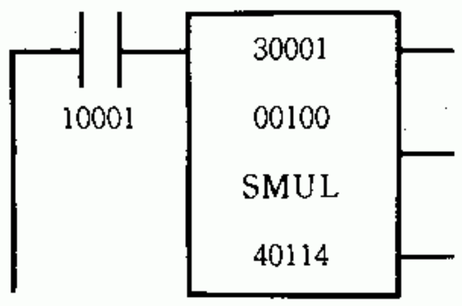
Example 3:



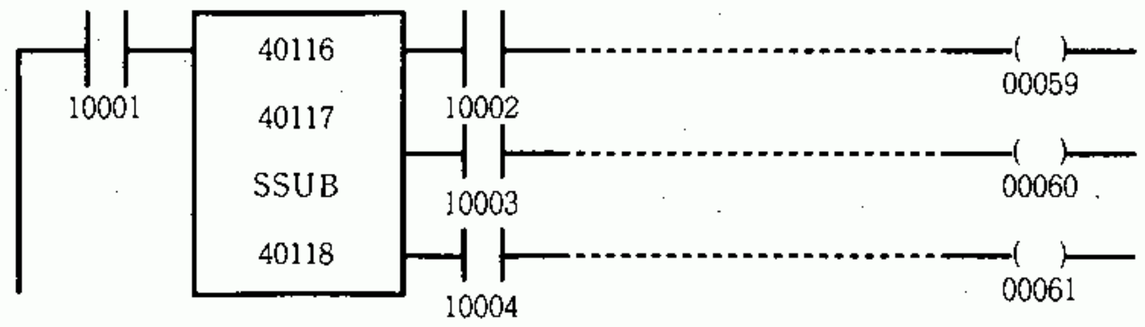
(3) Outputs of Signed Arithmetic Logic

Coils need not be connected to three output nodes (1, 2, and 3) of signed arithmetic function. A relay contact may be connected to the output nodes on the right or connect the output nodes directly to an input node of an arithmetic circuit, except relays.

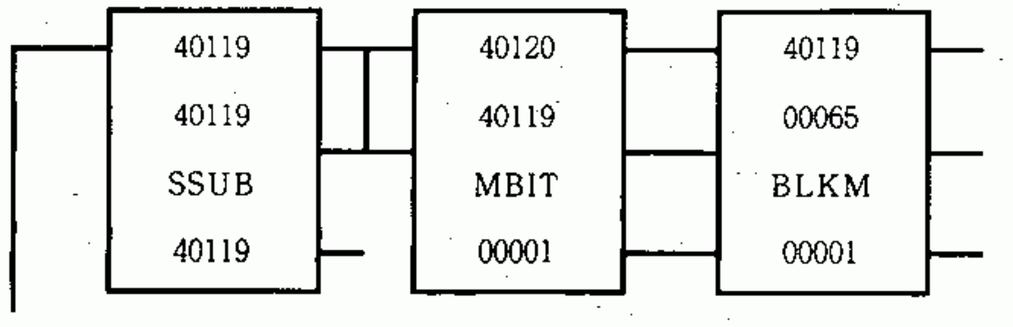
Example 1:



Example 2:

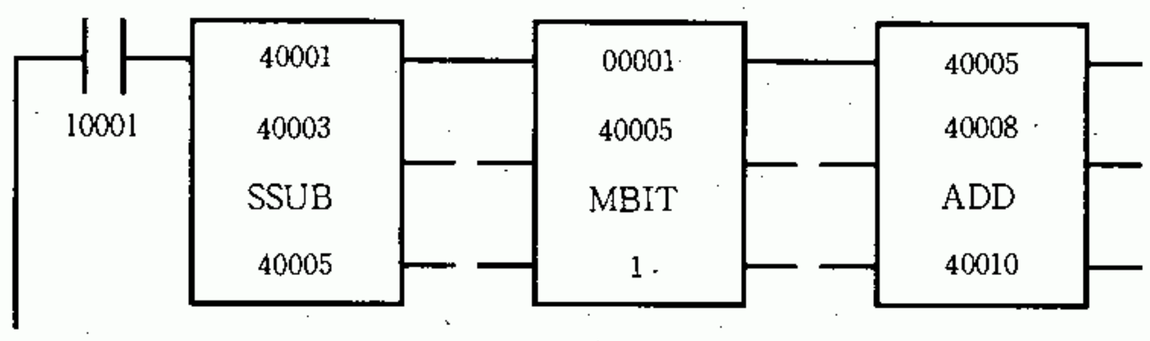


Example 3:



(4) Data Handling

In case the operation result of a signed arithmetic operation is used in an arithmetic operation, first obtain the absolute value of the operation result using "MBIT" as shown below, before using it in the next operation.
When the operation is executed with the negative number remaining as is, no correct result can be obtained. However, there will be no problem, even if an unsigned operation result is used in a signed arithmetic operation.

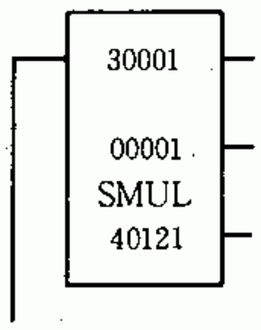


"1" in the MSB is cleared by MBIT.

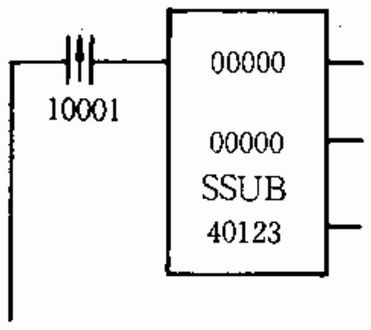
(5) Execution of Signed Arithmetic Operations (Only One Scanning Cycle)

To execute a constant arithmetic operation, connect the inputs directly to the power rail on the left. To execute it only in one scan, use a transitional contact as an input.

Example 1:



Example 2:



5.7 SQUARE ROOT

5.7.1 Types of Square Root

Square root is the function that calculates the square root of the operand V which is a 4- or 8-digit decimal number. Two types of square root are available as shown in Table 5.46

Table 5.46 Types of Square Root Function

Type	Symbol	Function	Range of Operand V	Reference Page
Square Root	SQRT	Calculation of $\sqrt{V}$	0 to 9,999	123
Double-precision Square Root	DSQR		0 to 99,999,999	125

5.7.2 Square Root (SQRT)

(1) Function

Operates square root in 4-digit decimal up to 4 places of decimal.

(2) Form

- Fig. 5.47 shows the form of square root (SQRT).

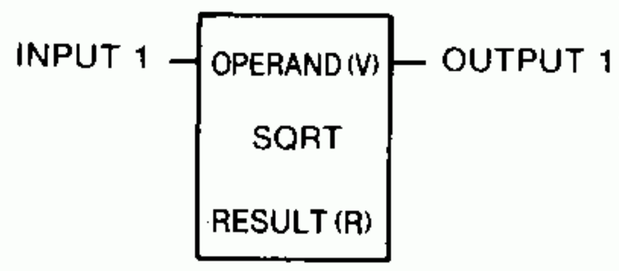


Fig. 5.47 SQRT General Form

- SQRT is the symbol denoting the square root.
- Square root operation requires two elements placed vertically (top and bottom). Referring to Table 5.47, specify any of reference numbers of various registers for each of the elements.

Table 5.47 Elements of Square Root

Element Position	Specified Number	Description
Top	Either one of the following : • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)	The contents of specified registers are the operand (V = 0 to 9,999).
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	The result of square root operation ($\sqrt{V}$) is stored as follows. × × × × × × × × × × + 1 Integer part of $\sqrt{V}$. Decimal part of $\sqrt{V}$

(3) Operation

By the square root (SQRT), $\sqrt{V}$ is calculated when the input 1 is ON. The result is treated as follows. The integer part of $\sqrt{V}$ is stored in R and the decimal part (rounded off at the fifth decimal place) in R+1. The output 1 is turned ON. The result remains in R and R+1 even after the input 1 is turned from ON to OFF.

Table 5.48 shows a square root operation (SQRT).

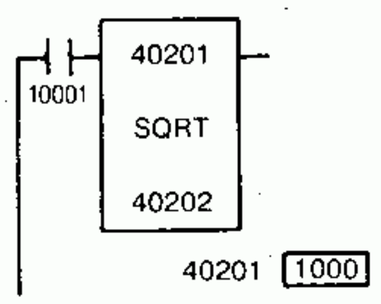
Table 5.48 SQRT Operation

Input 1	Operation	Output 1
ON	$\sqrt{V} \rightarrow$ R (Integer part), R+1 (Decimal part) *	ON
OFF	Not operated.	OFF

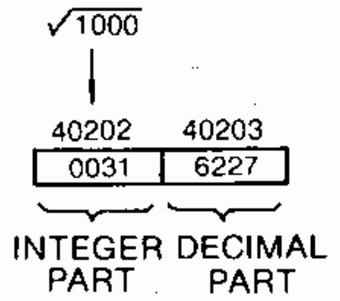
* Rounded off at the 5th decimal place.

(4) Example

Example:



(a) Ladder



(b) SQRT Operation

SQRT in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 is turned on. The result remains in 40202 and 40203 even after input relay 10001 is turned off.

5.7.3 Double-precision Square Root (DSQR)

(1) Function

Operates double-precision square root in 8-digit decimal up to 4 places of decimal.

(2) Form

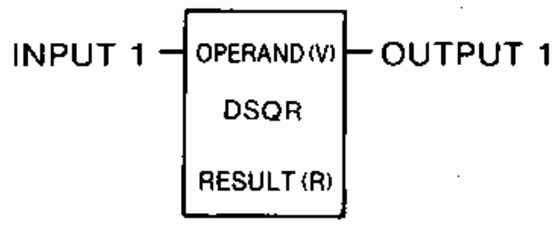


Fig. 5.48 DSQR General Form

- Fig. 5.48 shows the form of double-precision square root (DSQR).
- DSQR is the symbol denoting the double-precision square root.
- Double-precision square root operation requires two elements placed vertically (top and bottom). Referring to Table 5.49, specify either register reference number for each of the elements.

Table 5.49 Elements of DSQR Function

Element Position	Specified Number	Description
Top	Either one of the following : • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	The operand (V=0 to 99,999,999) is stored as tollows. ××××× ××××× +1 [V_H] [V_L] V_H: Higher-place 4 decimal digits of V V_L: Lower-place 4 decimal didits of V
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	The result of double-precision square root operation ($\sqrt{V}$) is stored as follows. ××××× ××××× +1 [ ] [ ] Integer part of $\sqrt{V}$ Decimal part of $\sqrt{V}$

(3) Operation

- By double-precision square root (DSQR), $\sqrt{V}$ is calculated when the input is ON. The result is treated as follows. The integer part of $\sqrt{V}$ is stored in R and the decimal part (rounded off at the fifth decimal place) in R+1. The output 1 is turned on.)
- The result remains in R and R + 1 even after the input 1 is turned from ON to OFF.
- Table 5.50 shows a double-precision square root (DSQR).

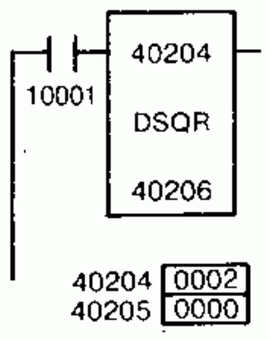
Table 5.50 Double-precision Square Root Operation

Input 1	Operation	Output 1
ON	$\sqrt{V} \rightarrow$ R (Integer part), R+1 (Decimal part)*	ON
OFF	Not operated.	OFF

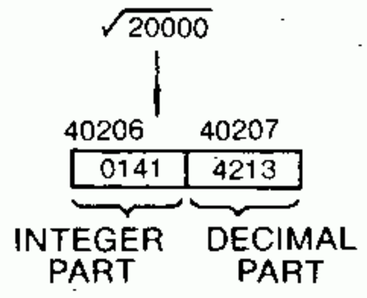
* Rounded off at the 5th decimal place.

(4) Example

Example:



(a) Ladder



(b) DSQR Operation

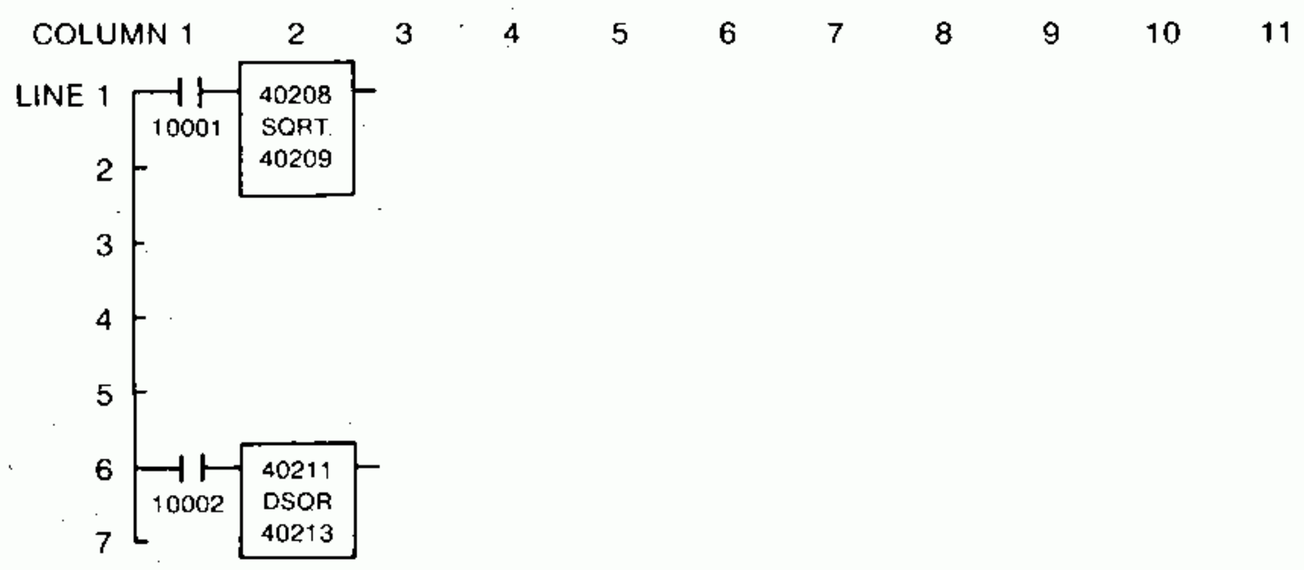
DSQR in (a) executes the operation of (b) when input relay 10001 is ON. The output 1 is turned on. The result remains in 40206 and 40207 even after input relay 10001 is turned off.

5.7.4 Programming Square Root Circuit and Precautions

(1) Programming Square Root Circuit

Square root operation requires two elements placed vertically (top and bottom). It can be used at any intersection of the 7 lines-by-10 columns matrix, but the top element (operand) cannot be used on line 7.

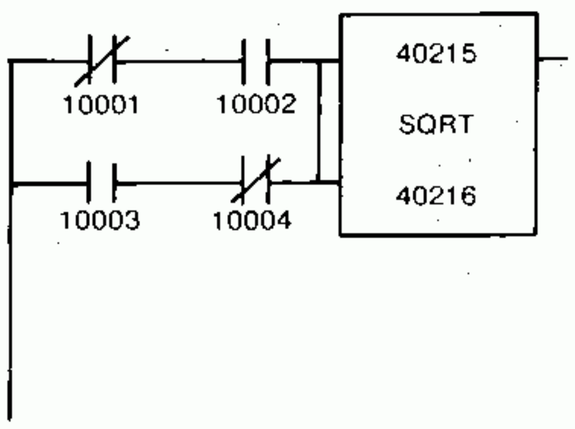
Example:



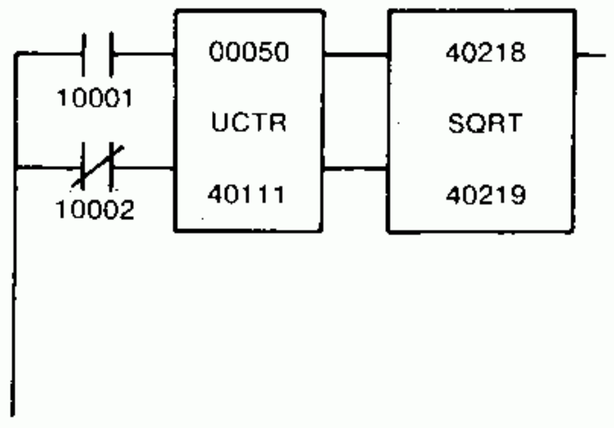
(2) Input of Square Root

Input to square root may be the output of other square roots, relays, timers, counters, or data processing circuits.

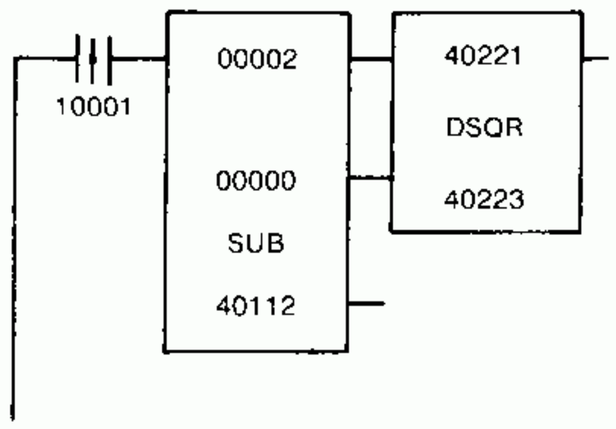
Example 1:



Example 2:



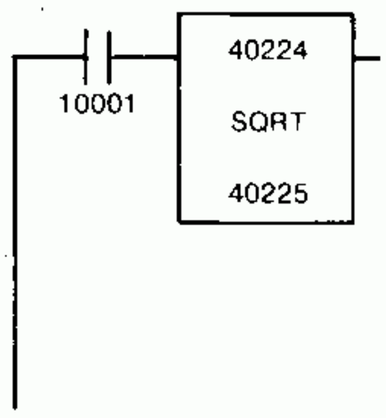
Example 3:



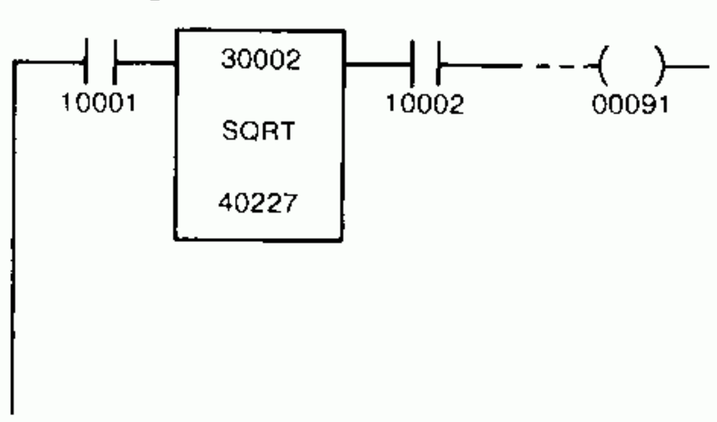
(3) Outputs of Square Root

Coils need not be connected to two output nodes (1 and 2) of a square root. It is permitted to connect a relay contact to the output node at right or connect the output nodes directly to an input node of an arithmetic circuit, except relays.

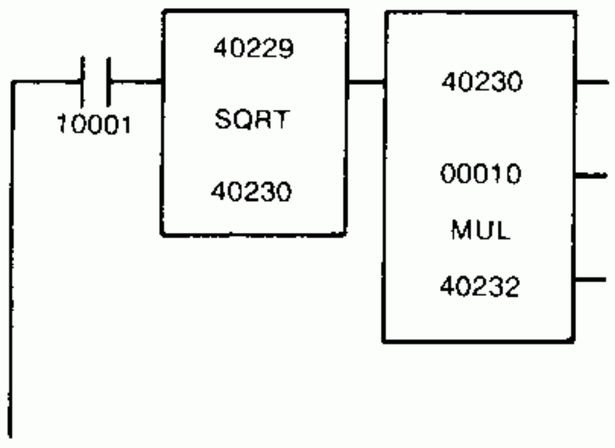
Example 1:



Example 2:



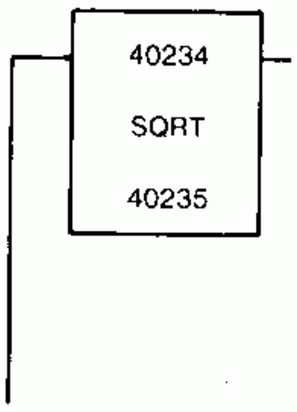
Example 3:



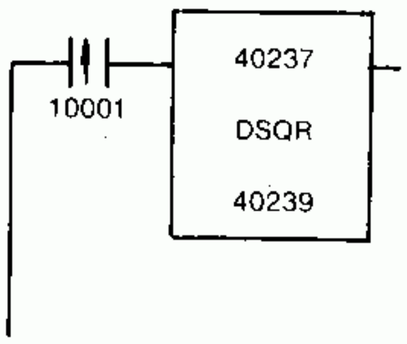
(4) Execution of Square Root (Only 1 Scan Cycle)

To execute a square root constantly, connect the inputs directly to the power rail at left. To execute it only during one scanning cycle, use a transitional contact as an input.

Example 1:



Example 2:



5.8 TRIGONOMETRIC FUNCTION

5.8.1 Types of Trigonometric Function Operations

Trigonometric function operation is intended to obtain the sine and cosine values of operand V down to the 4th decimal place in the range from 0° to 360°. There are two types of trigonometric function operations as shown in Table 5.51.

Table 5.51 Types of Trigonometric Function Operations

Type	Symbol	Content of Operation	Range of Operand V	Reference page
Sine	SIN	Calculation of SIN(V)	0.0000 ~ 360.0000	128
Cosine	COS	Calculation of COS(V)		130

Note The operands of trigonometric function are represented in degrees.
When any numeric value represented in radian is input, no correct result will be obtained.

5.8.2 Sine (SIN)

(1) Function

Obtains the value of the sine in the range form 0° to 360°.

(2) Form

- Fig. 5.49 shows the form of sine (SIN).
- SIN is the symbol denoting the sine.
- SIN operation requires two elements placed vertically (top and bottom). Referring to Table 5.52, specify any register reference number for each of the elements.

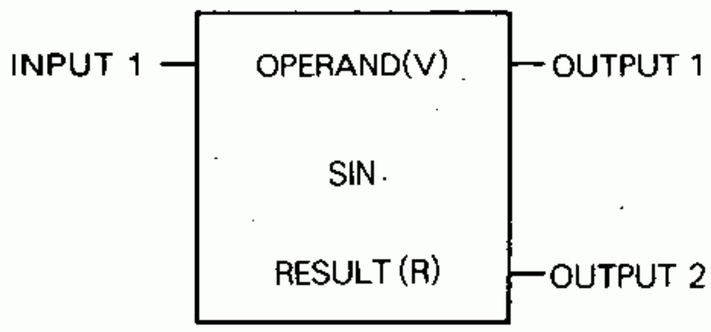


Fig. 5.49 SIN General Form

Table 5.52 Elements of SIN

Element Position	Specified Number	Description
Top	Either one of the following : • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	The contents of specified registers are the operand (V = 0.0000 to 360.0000). ×××××: 0××× Integer part ××××× + 1: ×××× Decimal part
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	The result of sine operation is stored as follows: ×××××: 000× Integer part ××××× + 1: ×××× Decimal part

(3) Operation

- When input 1 is ON, SIN will calculate SIN(V) and process the result as follows: It stores the integer of SIN(V) in R and the fraction (The fifth decimal place and greater are discarded.) in R+1. It turns output 1 to ON when the result is negative. Output 1 remains in the OFF state when the result is positive. It turns output 2 to ON when the value of V exceeds 360°, and no operation is executed. Even when input 1 turns from ON to OFF, the operation result remains in R and R+1, respectively.
- The actions of SIN are tabulated in Table 5.53.

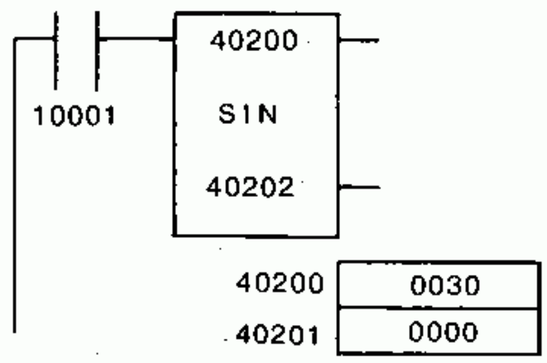
Table 5.53 Actions of SIN

Input 1	Operation	Result	Output 1	Output 2
ON	SIN (V) → R (Integer part)	Positive	OFF	OFF
	R+1 (Decimal part)*	Negative	ON	OFF
	If V > 360° †	-	OFF	ON
OFF	No operation.	-	OFF	OFF

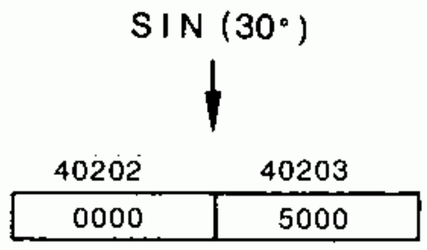
*The fifth decimal place and greater are discarded.
† No operation is performed even when input 1 is ON.

(4) Examples

Example 1:



(a) Ladder

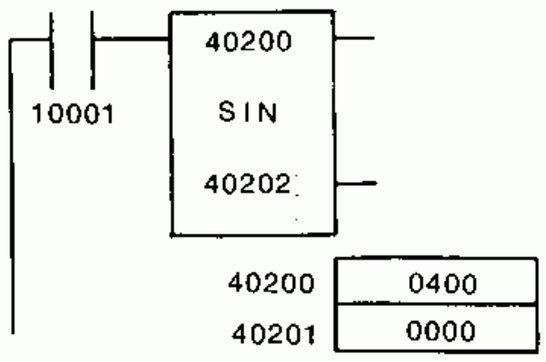


(b) Operation Content

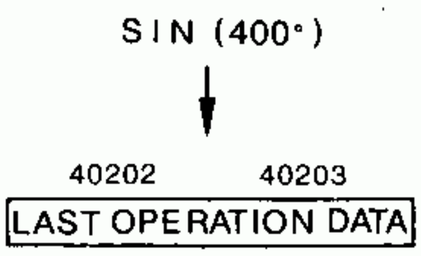
When relay 10001 is ON, SIN in (a) executes the operation of (b). Output 1 remains in the OFF state.

Even when input relay 10001 turns from ON to OFF, the operation result remains in 40202 and 40203, respectively.

Example 2:



(a) Ladder



(b) Operation Content

Since the value of V exceeds 360°, even when input relay 10001 turns to ON, the SIN in (a) turns output 2 to ON, without executing the operation.

5.8.3 Cosine (COS)

(1) Function

Obtains the value of cosine in the range from 0° to 360°.

(2) Form

- Fig. 5.50 shows the form of cosine (COS).

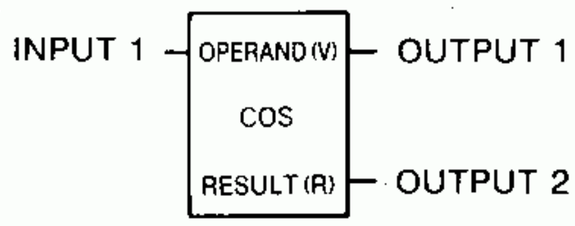


Fig. 5.50 COS General Form

- COS is the symbol denoting the cosine.
- COS operation requires two elements placed vertically (top and bottom). Referring to Table 5.54, specify register reference number for each of the elements.

Table 5.54 Elements of COS

Element Position	Specified Number	Description
Top	Either one of the following : • Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)	The contents of specified reference are the operand (V = 0.0000 to 360.0000). ×××××: 0××× (Integer part) ×××××+1: ×××× (Decimal part)
Bottom	• Holding register (40001-42047) • Link register (R0001-R1023)	The result of COS operation is stored as follows: ×××××: 000× (Integer part) ×××××+1: ×××× (Decimal part)

(3) Operation

- When input 1 is ON, COS will calculate COS(V) and process the result as follows: It stores the integer of COS (V) in R and the fraction (The fifth decimal place and greater are discarded.) in R+1. It turns output 1 to ON when the result is negative. Output 1 remains in the OFF state when the result is positive. It turns output 2 to ON when the value of V exceeds 360°, and no operation is executed. Even when input 1 turns from ON to OFF, the operation result remains in R and R+1, respectively.
- The actions of COS are tabulated in Table 5.55.

Table 5.55 Actions of COS

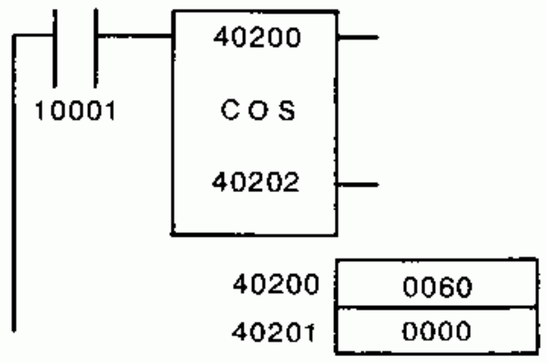
Input 1	Operation	Result	Output 1	Output 2
ON	SIN (V) → R (Integer part) R+1 (Decimal part)*	Positive	OFF	OFF
		Negative	ON	OFF
	If V > 360° †	-	OFF	ON
OFF	No operation	-	OFF	OFF

* The fifth decimal place and greater are discarded.

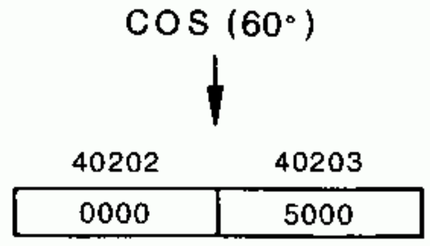
† No operation is performed even when input 1 is ON.

(4) Examples

Example 1:



(a) Ladder

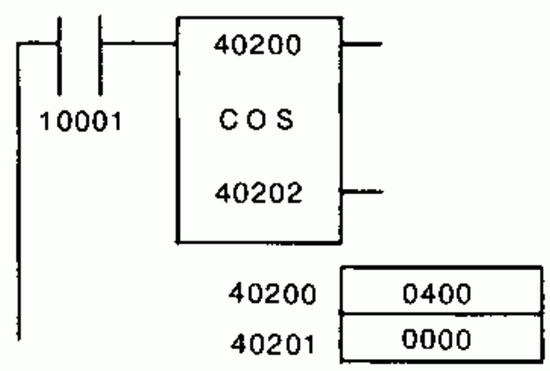


(b) Operation Content

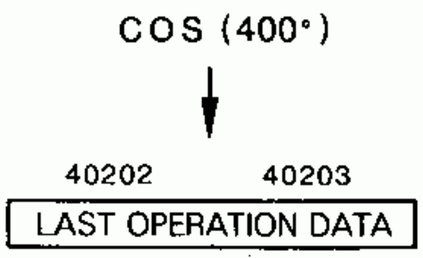
When relay 10001 is ON, SIN in (a) executes the operation of (b). Output 1 remains in the OFF state.

Even when input relay 10001 turns from ON to OFF, the operation result remains in 40202 and 40203, respectively.

Example 2:



(a) Ladder



(b) Operation Content

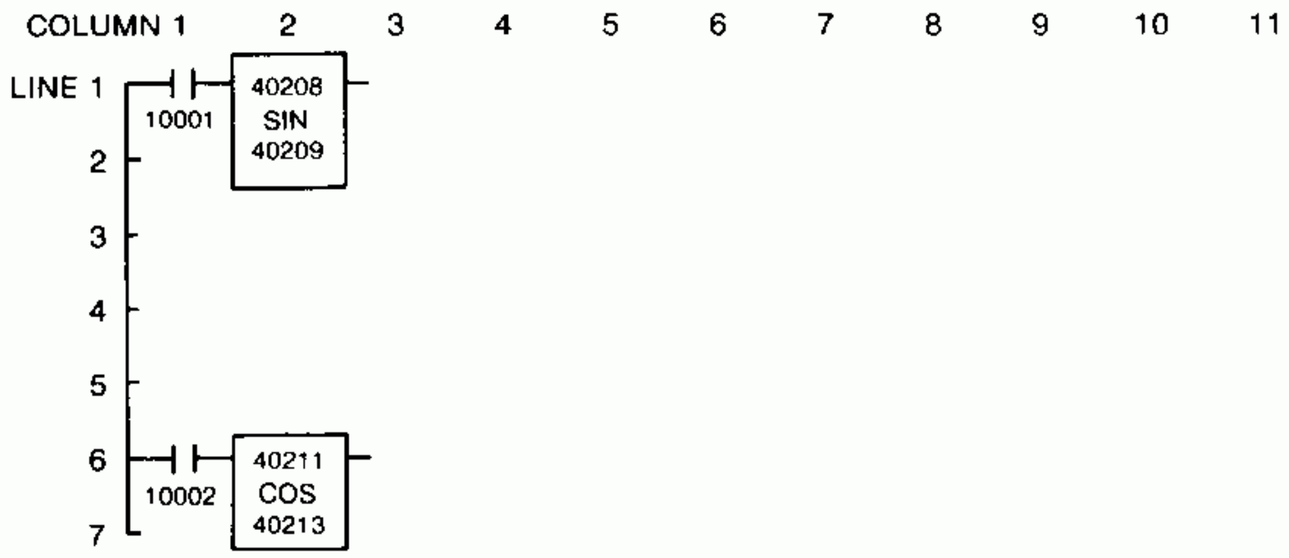
Since the value of V exceeds 360°, even when input relay 10001 turns to ON, the COS in (a) turns output 2 to ON, without executing the operation.

5.8.4 Programming Trigonometric Function Circuit and Precautions

(1) Programming Trigonometric Function Circuit

Trigonometric function requires two elements placed vertically (top and bottom). It can be used at any intersection of the 7 lines-by-10 columns matrix, but the top element (operand) cannot be used on line 7.

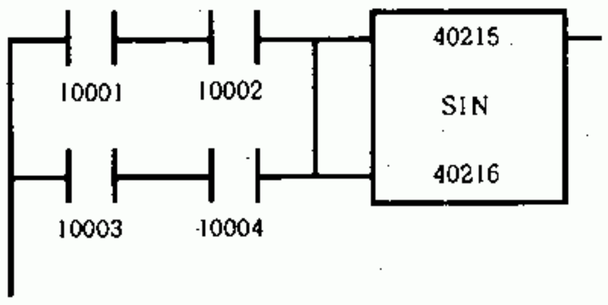
Example:



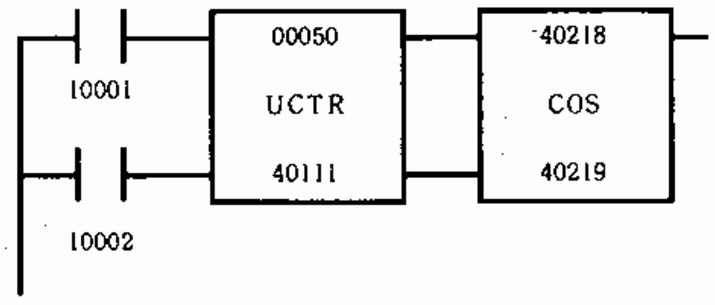
(2) Input of Trigonometric Function

Input to trigonometric function may be the output of other square roots, relays, timers, counters, or data processing circuits.

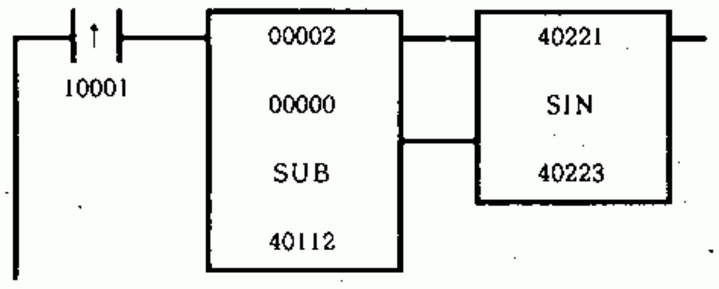
Example 1:



Example 2:

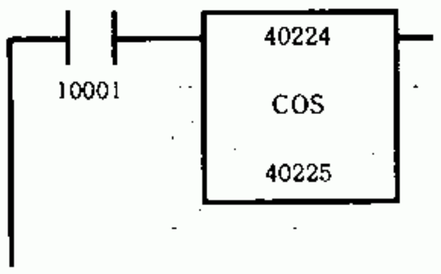


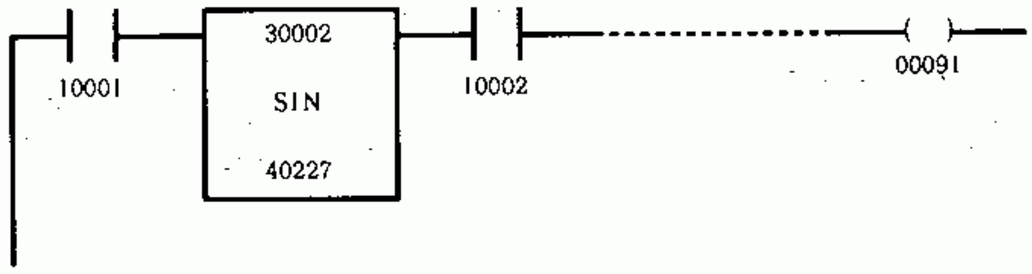
Example 3:



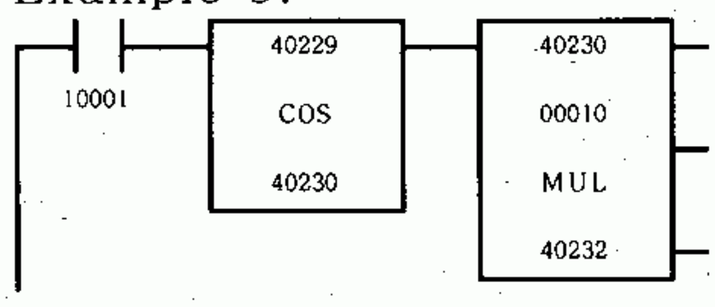
(3) Outputs of Trigonometric Function

Coils need not be connected to two output nodes (1 and 2) of trigonometric function. It is permitted to connect a relay contact to the output node at right or connect the output nodes directly to an input node of an arithmetic circuit, except relays. Output node 1 is ON only when result of operation is negative.





Example 3:



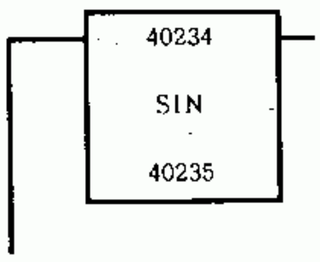
(4) Unit of Operand

The unit of operands handled in the operation of trigonometric functions is degrees. When any angle represented in radian is used as data, no correct result can be obtained, because the operation is performed according to the degree system.

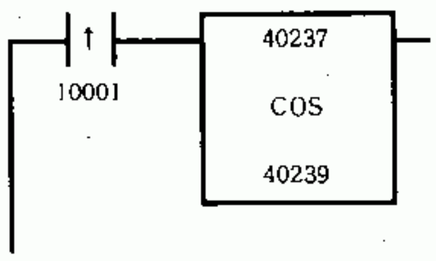
(5) Execution of Trigonometric Function (Only 1 Scan Cycle)

To execute a trigonometric functions constantly, connect the inputs directly to the power rail at left. To execute it only during one scanning cycle, use a transitional contact as an input.

Example 1:



Example 2:



5.9 DATA MOVE

Numerical values in data tables can be transferred or moved to other data tables within the controller with this function. This function has the ability to simulate the operation of electro-mechanical relays, timers and counters, as well as to perform arithmetic functions (addition, subtraction, multiply, and divide), and square root functions.

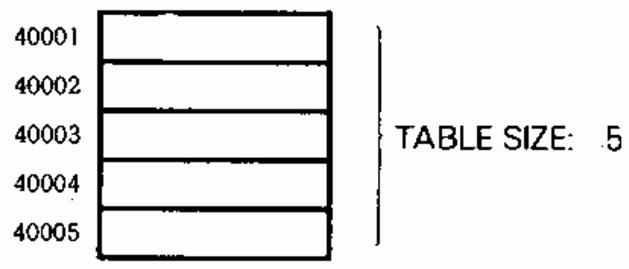
5.9.1 Basic Terminology

(1) Data Table and Table Size

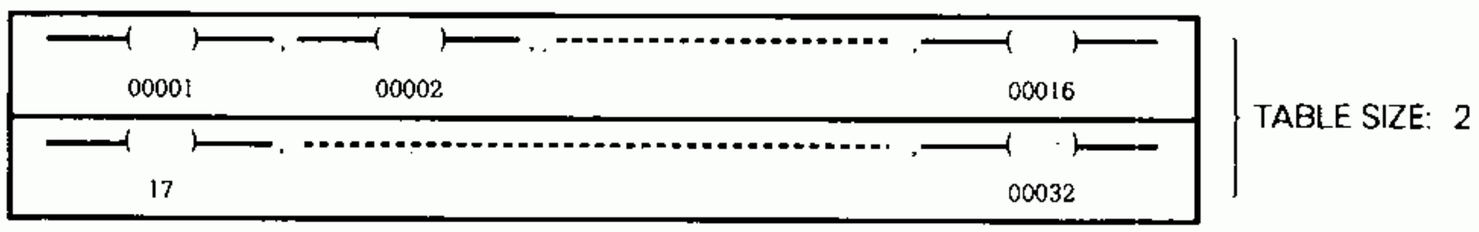
The term "Data Table" refers to register tables, coil tables and relay tables having serial numbers.

The term "Table size" referes to the size of data table, and it is counted in register units in the case of register tables and in hexadecimal units in the cases of coil tables and relay tables.

Example 1: Data table consisting of five serial holding registers



Example 2: Coil table consisting of 32 continuous coils



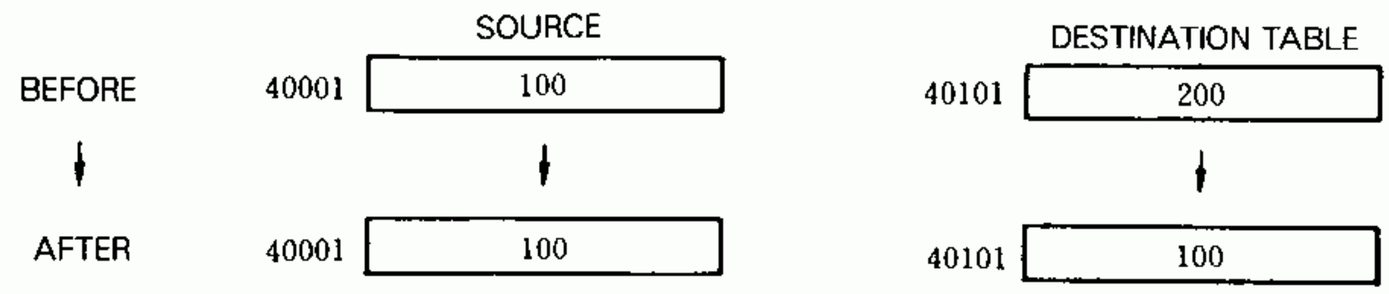
(2) Source and Destination

The source of data transfer is called "Source" and the destination of data transfer is called "Destination". Data must be transferred between the source and the destination.

① The content of the source is stored in the destination.

② The content of the source is the same as that before execution of transfer.

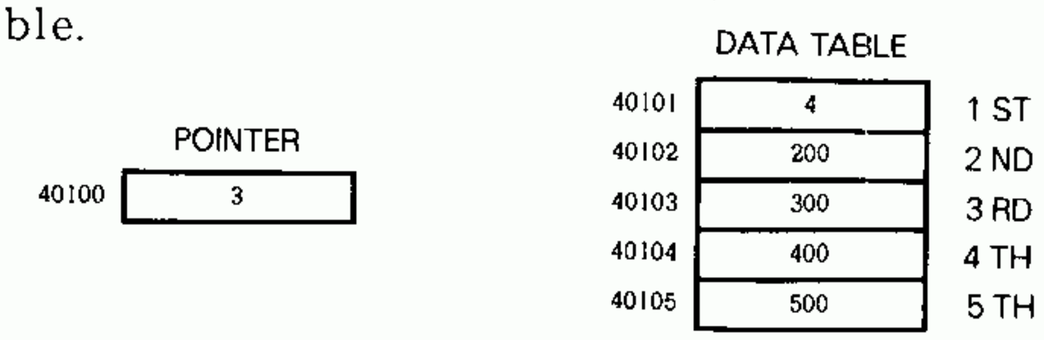
Example:



(3) Pointer

The term "Pointer" refers to the register used to indicate a specified position in data table.

Example:



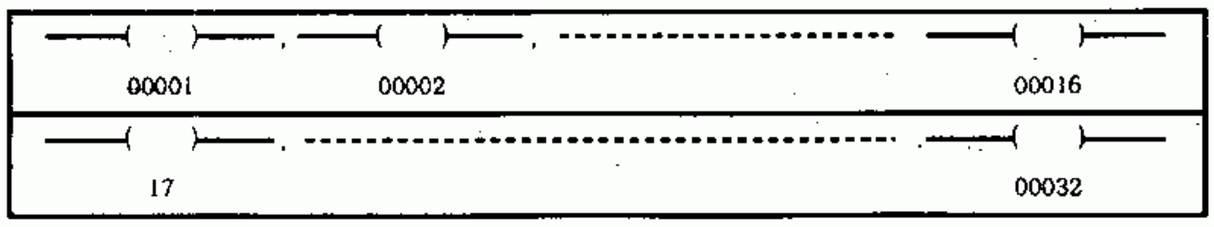
In the example shown in the preceding page, where the content of the pointer is 3, the pointer indicates 40103 located at the third place from the top.

Note that the pointer may be on the source side or on the destination side.

(4) Reference Numbers of Source and Destination

① In case the reference number of coil or relay is specified as the reference number of source or destination, it is necessary that the relation of n = 16m + 00001 (m = 0, 1, 2 ...) holds for coil or n = 16m + 10001 (m = 0, 1, 2 ...) holds for input relay, where the lower-place 4 digits is assumed to be n.

Example:



The example shown above is a case where m = 0, and the reference No. (n) that is specified is 00001.

② In the case of coil or relay, even if the actual need is less than 16 out of a set of 16 points and the rest are not used, 16 points will be transferred in one lot.

③ In case the destination are coils, the number of the coil table cannot overlap the coil No. (even one) used in another circuit.

5.9.2 Types of Data Move

Nine types of data move are available, as shown in Table 5.56.

Table 5.56 Types of Data Move

Type	Symbol	Reference Page
Block Move	BLKM	135
Register-to-Table Move	R → T	138
Table-to-Register Move	T → R	141
Table-to-Table Move	T → T	144
First In	FIN	147
First Out	FOUT	149
Table Search	SRCH	153
Table Set	TSET	156
Get Controller System Status	STAT	158

5.9.3 Block Move (BLKM)

(1) Function

This is a move from a source table to a destination table. This function causes an entire table of registers to be copied into another table in one scan.

(2) Form

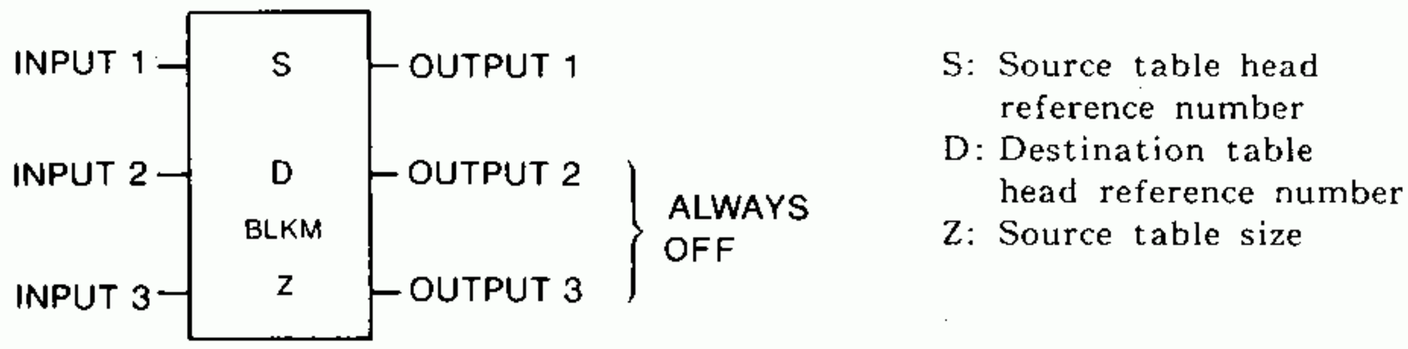


Fig. 5.51 BLKM General Form

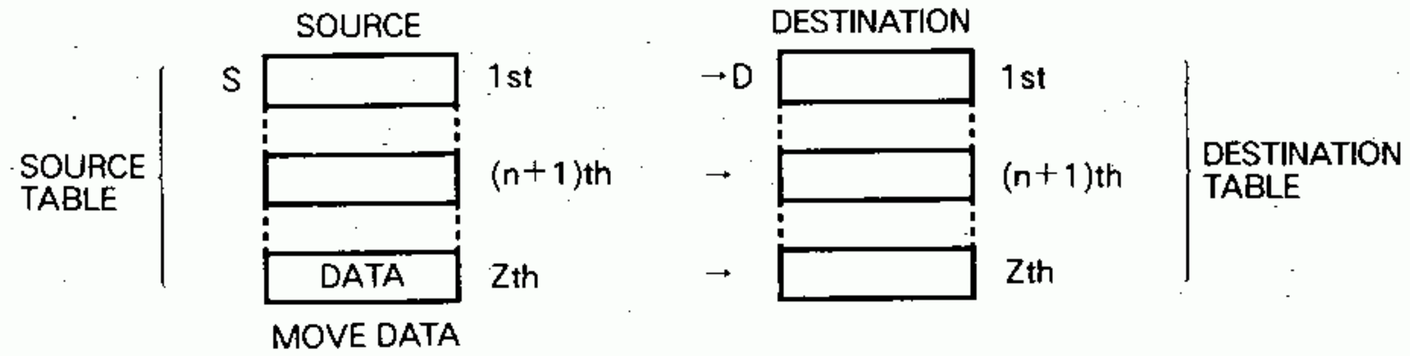
- Fig. 5.51 shows the form of block move.
- BLKM is the symbol denoting the block move.
- BLKM operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.57, specify the needed number for each element.

Table 5.57 Elements of BLKM

Element	Description	Specified Number	
Top	Source table head reference number	• Coil	(00001-02033)
		• Input relay	(10001-10497)
		• Link coil	(D0001-D1009)
		• Input register	(30001-30128)
		• Holding register	(40001-42048)
		• Link register	(R0001-R1024)
Middle	Destination table head reference number	• Coil	(00001-02033)
		• Link coil	(D0001-D1009)
		• Holding register	(40001-42048)
		• Link register	(R0001-R1024)
Bottom	Table size	• Coil	(1-100)
		• Input relay	(1-32)
		• Link coil	(1-64)
		• Each register	(1-100)

Note
1. When a relay reference No. is to be specified in the top stage, the lower - place 4 digits of the number that can be specified will be limited to [16 m + 1]. (m = 0, 1, 2, ...)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.

(3) Operation



Note All the registers (discrete) will be moved in one scanning cycle.

(a) Inputs

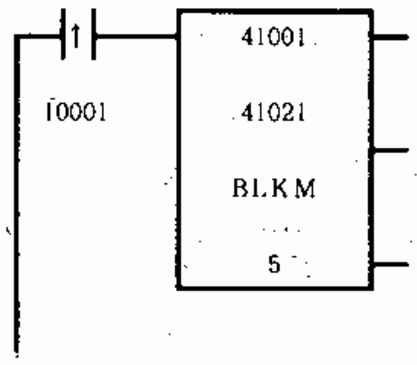
Only the input 1 is used with the BLKM function. When this input node receives power flow, the move is performed. Every scan, when enabled, the Block move will operate upon all registers.

(b) Outputs

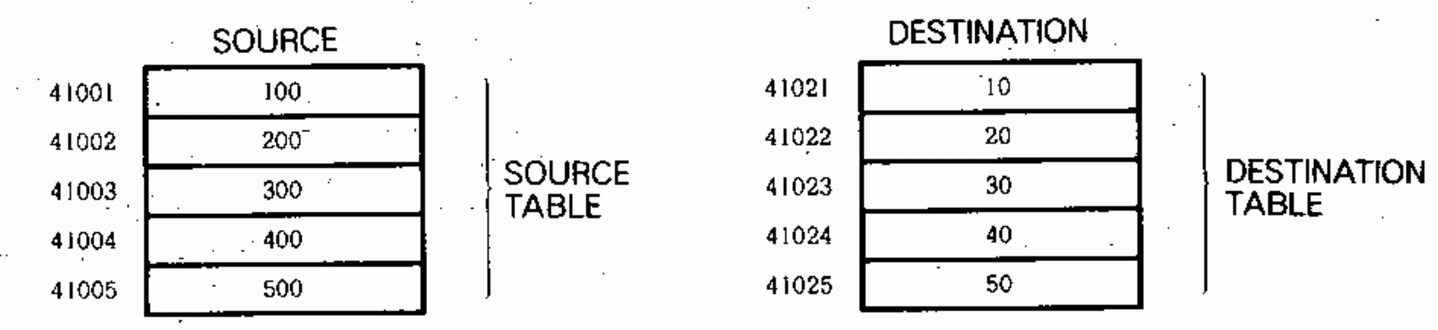
The block move function utilizes only the output 1. The lower two outputs 2 and 3 have no significance and will be OFF under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network.

(4) Example

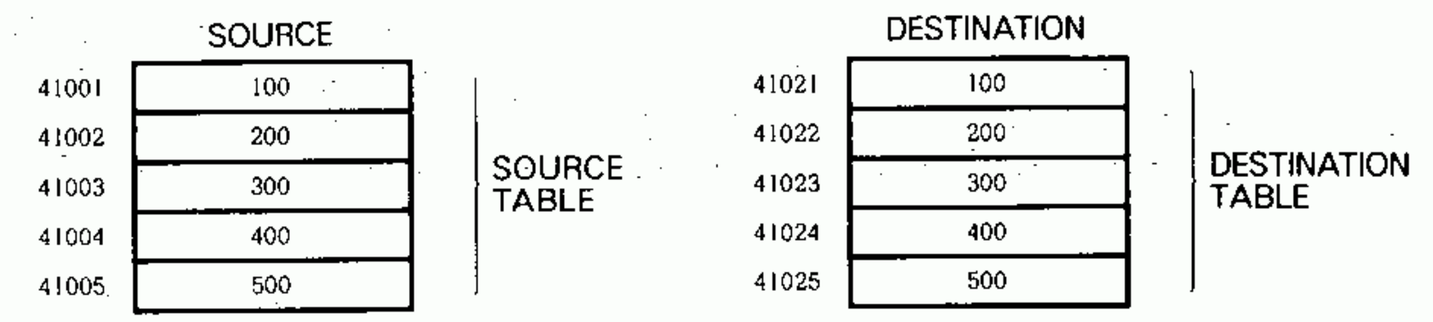
Example 1:



(a) Ladder



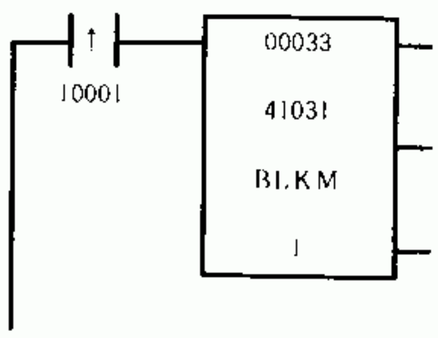
① Registers before Move



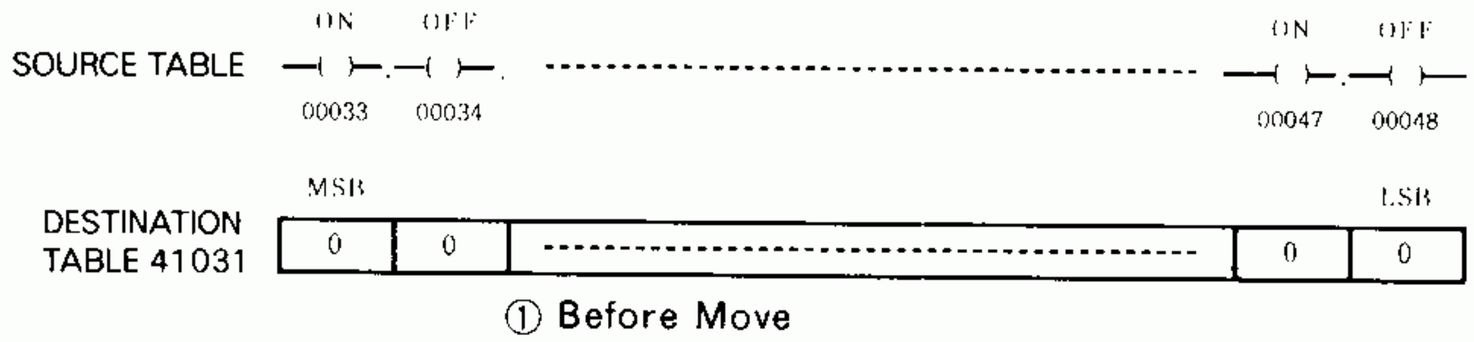
② Registers after Move

(b) BLKM Operation

Example 2:



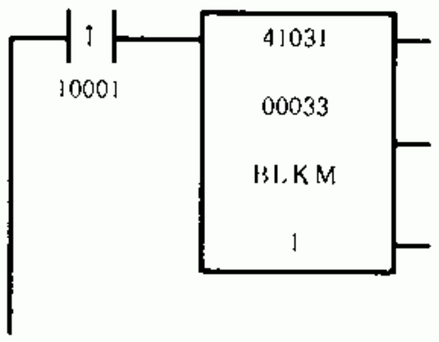
(a) Ladder



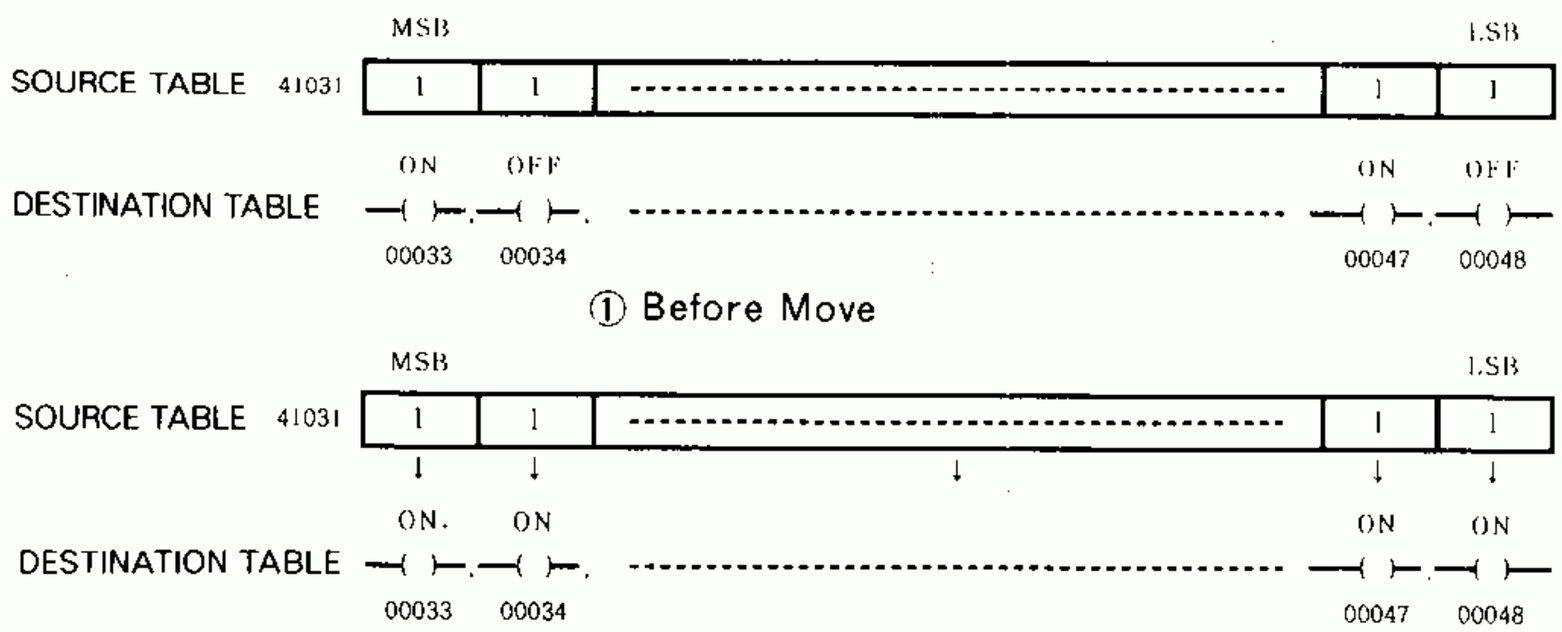
(b) BLKM Operation

If the 16 points of source coil 00033 and onward are transferred to destination 41031, each of the ON and OFF states of the 16 points of coil 00033, 00034, ……, 00047, 00048 will enter the corresponding bits of 41031, with the ON state in the form of "1" and with the OFF state in the form of "0".

Example 3:



(a) Ladder



(b) BLKM Operation

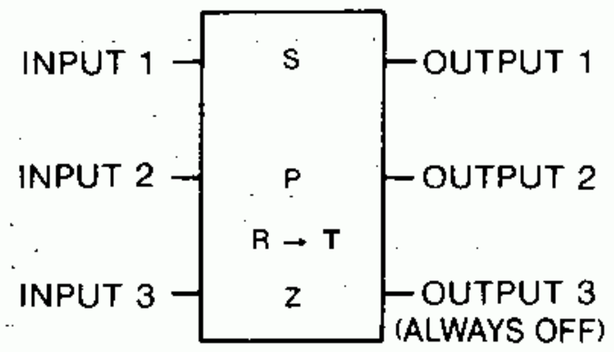
If the source holding register 41031 is transferred to the 16 points of destination coil 00033 and onward, the state of "1" and "0" of each bit of 41031 will be set in the state of "ON" and "OFF", respectively, in the 16 coils of 00033, 00034, ... 00047, 00048.

5.9.4 Register–to–Table Move (R → T)

(1) Function

This is a move from a source to a destination table.

(2) Form



S: Source table head reference number
P: Pointer
Z: Destination table size

Fig. 5.52 R → T General Form

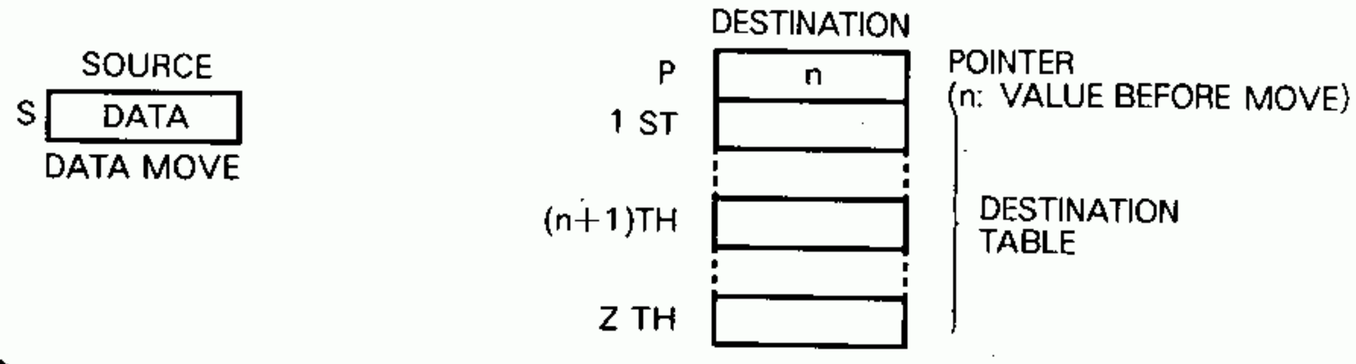
- Fig. 5.52 shows the form of register-to-table move.
- R → T is the symbol denoting the register-to-table move.
- R → T operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.58, specify the needed number for each element.

Table 5.58 Elements of R → T

Element	Description	Specified Number
Top	Source reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Pointer	• Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-999)

Note
1. When a relay reference No. is to be specified in the top stage, the lower - place 4 digits of the number that can be specified will be limited to [16 m + 1]. (m = 0, 1, 2, ...)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.
3. Destination table starts with the reference No. next to the pointer.
4. Pointer is not included in the table size.

(3) Operation



Note Move will be performed at the rate of one register (or a set of 16 discrete points) per scanning cycle.

(a) Inputs

All three inputs are used with the R → T function. The input 1 controls the move. Every scan power flow is received at this node, the move is performed and the pointer incremented. Both the move and pointer incrementing occur during the solution of this Function Block. Incrementing the pointer will cause moves on future scans to occur into successive register locations. The pointer cannot exceed the table length. Thus when the pointer is equal to the table length, it will stop incrementing and the move will stop operating. A transitional contact can be used to control the input 1 if a single move operation is desired.

The input 2, when receiving power flow, inhibits the incrementing of the pointer. Thus moves with power flow at both inputs 1 and 2 can be made continuously to the same register in the table, until either another function increments the pointer or the input 2 loses power flow. The input 3 resets the pointer to zero. Whenever this input receives power flow, the pointer is reset to zero regardless of its current value.

The bottom input, when energized, will cause the R → T move to go to the first register in the table if the top element is also energized.

(b) Outputs

The register-to-table function utilizes only the first two outputs. The output 3 has no significance and will be OFF (no power flow) under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chanied horizontally within a network. The output 2 supplies power flow whenever the pointer is equal to the table length, when the move function has reached the end of the table.

R → T move functions at I/O ON are shown in Tables 5.59 and 5.60.

Table 5.59 R → T Move Functions at Input ON

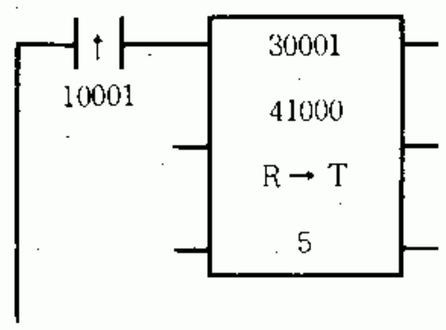
Input ×	Functions
Input 1	Executes move and adds 1 to the pointer.*
Input 2	Prohibits pointer remake.
Input 3	Sets the pointer to 0. (Regardless of Input 1 ON/OFF staus.)

* If pointer ≧ table size, move is not executed.

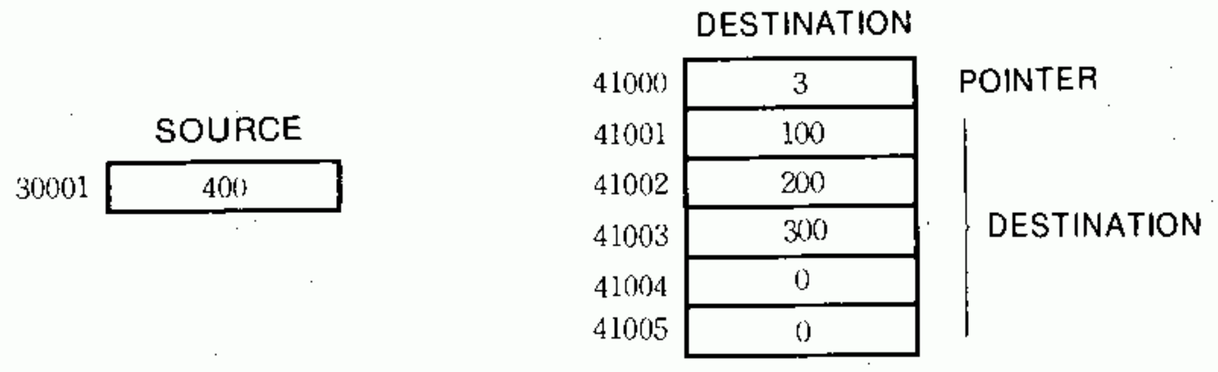
Table 5.60 Output ON Meaning (R → T Move)

Output ×	Meaning
Output1	Same as Input 1 ON/OFF status
Output2	Pointer value = Table size turns ON (Regardless of Input 1 ON/OFF Status.)
Output3	Always OFF

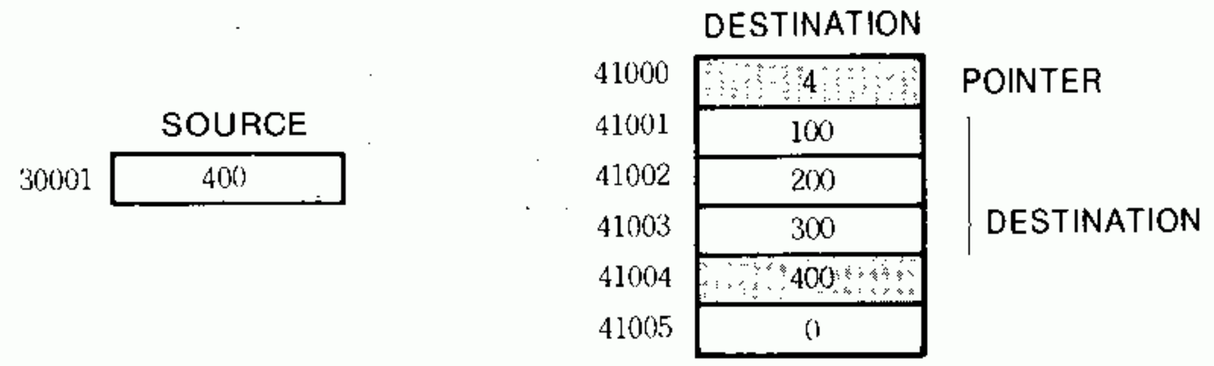
(4) Example



(a) Ladder



① Before Move



② After Move

(b) R → T Operation

R → T shown in (a) will execute transfer of data shown in (b), when input relay 10001 turns from OFF to ON. At this time, output 1 only will turn ON.

5.9.5 Table-to-Register Move (T → R)

(1) Function

This is the opposite function to the R → T, namely, a move from a source table to a destination.

(2) Form

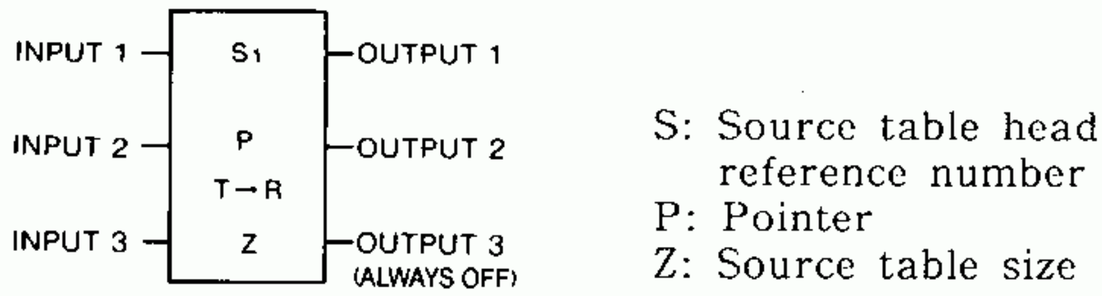


Fig. 5.53 T → R General Form

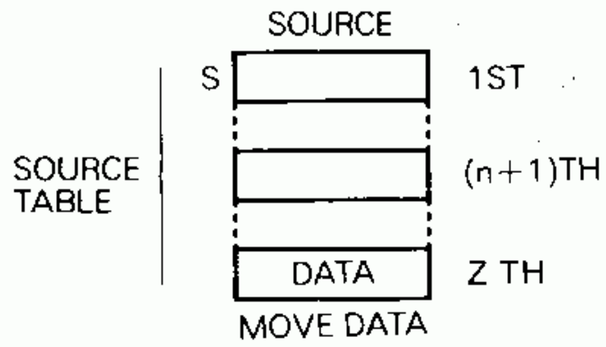
- Fig. 5.53 shows the form of block move.
- T → R is the symbol denoting the table-to-register move.
- T → R operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.61, specify the needed number for each element.

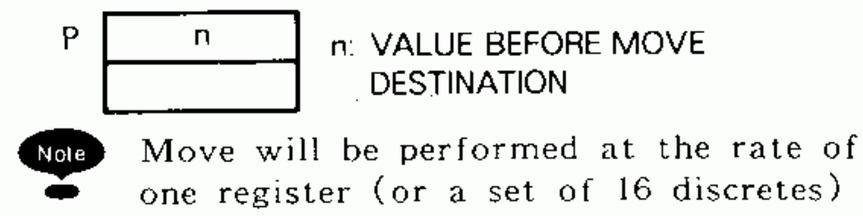
Table 5.61 Elements of T → R

Element	Description	Specified Number	
Top	Source reference number	• Coil	(00001-02033)
		• Input relay	(10001-10497)
		• Link coil	(D0001-D1009)
		• Input register	(30001-30128)
		• Holding register	(40001-42048)
		• Link register	(R0001-R1024)
Middle	Pointer	• Holding register	(40001-42047)
		• Link register	(R0001-R1023)
Bottom	Table size	• Coil	(1-128)
		• Input relay	(1-32)
		• Link coil	(1-64)
		• Input register	(1-128)
		• Holding register	(1-999)
		• Link register	(1-999)

Note
1. When a relay reference No. is to be specified in the top stage, the lower-place 4 digits of the number that can be specified will be limited to 〔16 m + 1〕. (m = 0, 1, 2, . . .)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.
3. Destination starts with the reference No. next to the pointer.

(3) Operation





(a) Inputs

All three inputs are used with the T → R function. The input 1 controls the move. Every scan power flow is received at this node, the move is performed and the pointer incremented. Both the move and pointer incrementing occur during the solution of this Function Block. Incrementing the pointer will cause moves on future scans to occur from successive register locations. The pointer can not exceed the table length. Thus when the pointer is equal to the table length, it will stop incrementing and the move will stop operating. A transitional contact can be used to control the input 1 if a single move operation is desired.

The input 2, when receving power flow, prohibits the incrementing of the pointer. Thus moves with power flow at both inputs 1 and 2 can be made continuously from the same register in the table until either another function increments the pointer or the input 1 loses power flow.

The input 3 resets the pointer to zero. Whenever this input receives power flow, the pointer is reset to zero regardless of its current value. The input 1, when energized at the same time as the input 3 will cause the first element in the table to be moved into the destination register.

(b) Outputs

The table-to-register function utilizes only the first two outputs. The output 3 has no significance and will be OFF (no power flow) under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever the pointer is equal to the table length, when the move function has reached the end of the table.

T → R move functions at I/O ON are shown in Tables 5.62 and 5.63.

Table 5.62 T → R Move Functions at Input ON

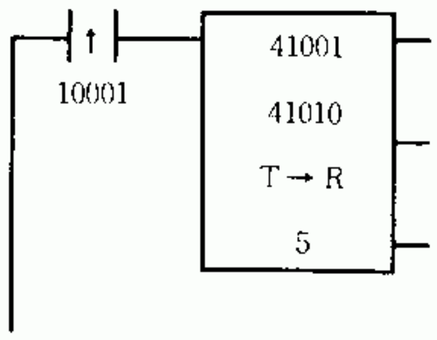
Input ×	Functions
Input 1	Executes move and add 1 to the pointer.*
Input 2	Prohibits pointer remake.
Input 3	Set the pointer to 0. (Regardless to Input 1 ON/OFF status.)

* If pointer ≧ table size, move is not executed.

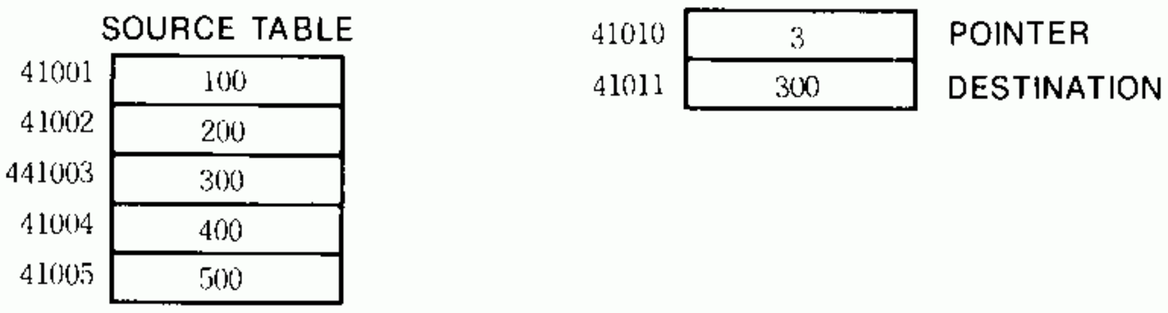
Table 5.63 Output ON Meaning (T → R Move)

Output ×	Meaning
Output 1	Same as Input 1 ON/OFF status.
Output 2	Pointer value = Table size turns ON(Regardless to Input 1 ON/OFF status.)
Output 3	Always OFF

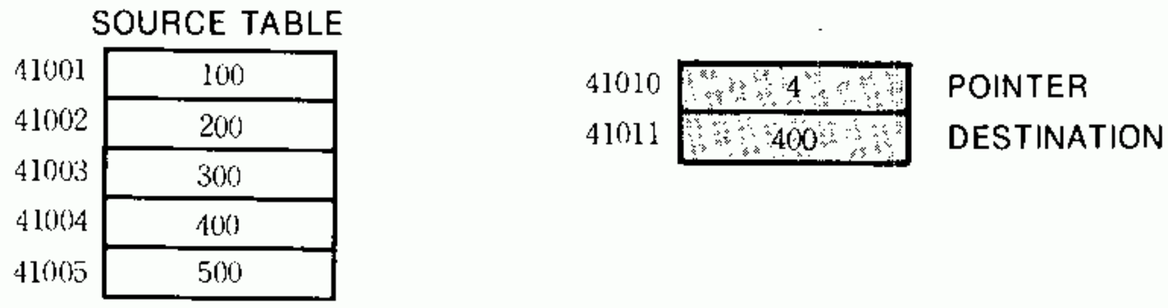
(4) Example



(a) Ladder



① Before Move



② After Move

(b) T → R Operation

T → R shown in (a) will execute transfer of data shown in (b) when input relay 10001 turns from OFF to ON. At this time, output 1 only will turn to ON.

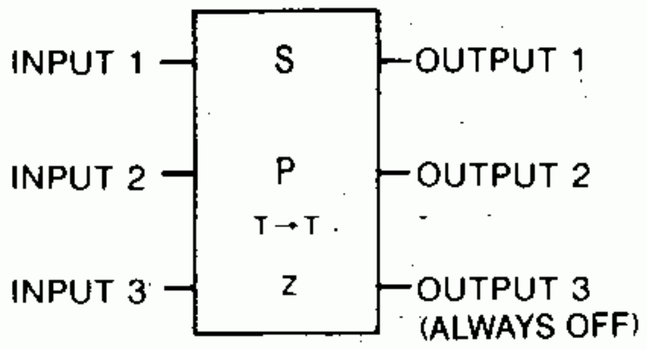
Unless holding register 41011 is not updated by another ladder, the relation between the pointer and destination indicates where the data of the holding register as a result of T → R is located in the table.

5.9.6 Table-to-Table Move (T → T)

(1) Function

This is a move function from a source table to a destination table.

(2) Form



S: Source table head reference number
P: Pointer
Z: Source table size

Fig. 5.54 T → T General Form

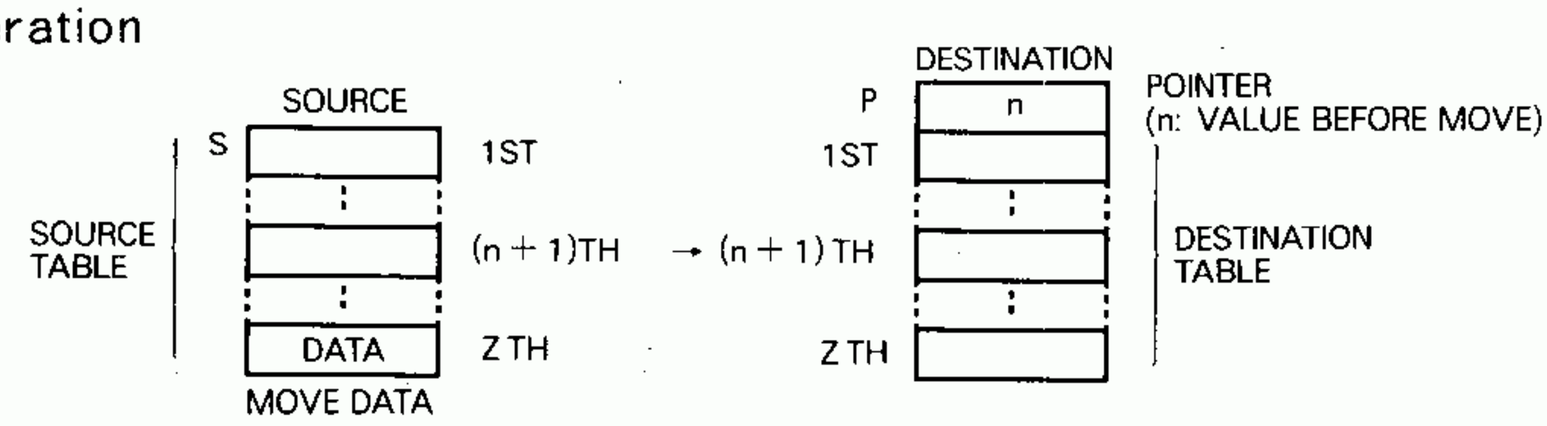
- Fig. 5.54 shows the form of table-to-table.
- T → T is the symbol denoting the table-to-table move.
- T → T operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.64, specify the needed number for each element.

Table 5.64 Elements of T → T

Element	Description	Specified Number	
Top	Source reference number	• Coil	(00001-02033)
		• Input relay	(10001-10497)
		• Link coil	(D0001-D1009)
		• Input register	(30001-30128)
		• Holding register	(40001-42048)
		• Link register	(R0001-R1024)
Middle	Pointer	• Holding register	(40001-42047)
		• Link register	(R0001-R1023)
Bottom	Table size	• Coil	(1-128)
		• Input relay	(1-32)
		• Link coil	(1-64)
		• Input register	(1-128)
		• Holding register	(1-999)
		• Link register	(1-999)

Note
1. When a relay reference No. is to be specified in the top stage, the lower- place 4 digits of the number that can be specified will be limited to 〔16 m + 1〕. (m = 0, 1, 2, . . .)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.
3. Destination starts with the reference No. next to the pointer.
4. Pointer is not included in the table size.

(3) Operation



Note Move will be performed at the rate of one register (or a set of 16 discretes) per scanning cycle.

(a) Inputs

All three inputs are used with the T → T function. The input 1 controls the move. Every scan power flow is received at this node, the move is performed and the pointer incremented. Both the move and pointer incrementing occur during the solution of this function Block. Incrementing the pointer will cause moves on future scans to occur at successive register locations. The pointer cannot exceed the table length. Thus when the pointer is equal to the table length, it will stop incrementing and the move will stop operating.

A transitional contact will be used to control the input 1 if the single move operation is desired.

The input 2, when receiving power flow at both inputs 1 and 2 can be made continuously between the same registers in the tables, until another function increments the pointer or the input 2 loses power flow. The output 3 can reset the pointer to zero. Whenever this input receives power flow, the pointer is reset to zero regardless of its current value.

The input 1, when energized at the same time as the input 3, will cause the first element in the source table to be copied to the first element in the destination table.

(b) Outputs

The table-to-table function utilizes only the first two outputs. The output 3 has no significance and will be OFF (no power flow) under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Funciton Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever the pointer is equal to the table length and indicates when the move function has reached the end of the table.

T → T move functions at I/O ON are shown in Tables 5.65 and 5.66.

Table 5.65 T → T Move Functions at Input ON

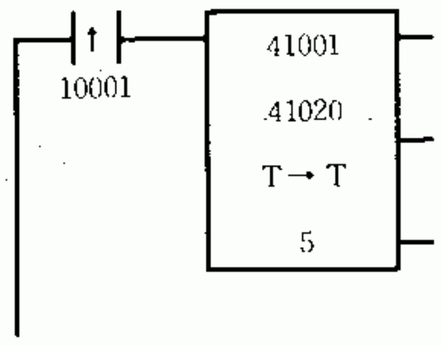
Input ×	Functions
Input 1	Executes move and add 1 to the pointer.*
Input 2	Prohibits pointer remake.
Input 3	Set the pointer to 0. (Regardless to Input 1 ON/OFF status.)

*If pointer ≧ table size, move is not executed.

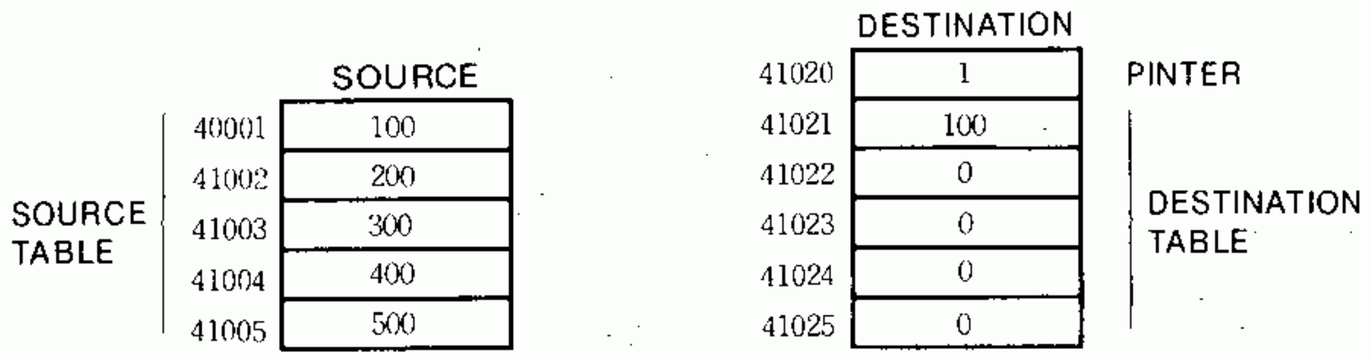
Table 5.66 Output ON Meaning (T → T Move)

Output ×	Meaning
Output 1	Same as Input 1 ON/OFF status
Output 2	Pointer value = Table size turns ON (Regardless to Input 1 ON/OFF Status.)
Output 3	Always OFF

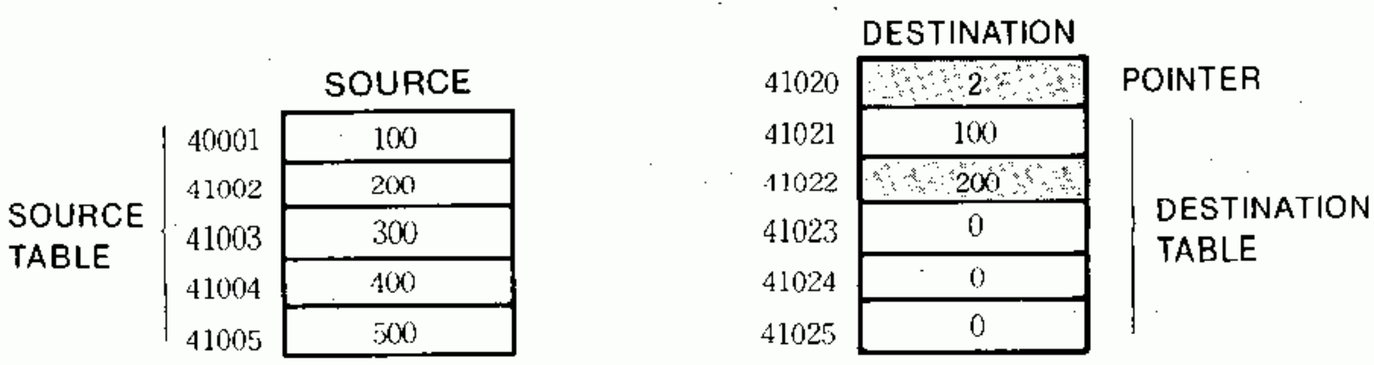
(4) Example



(a) Ladder



① Before Move



② After Move

(b) T → T Operation

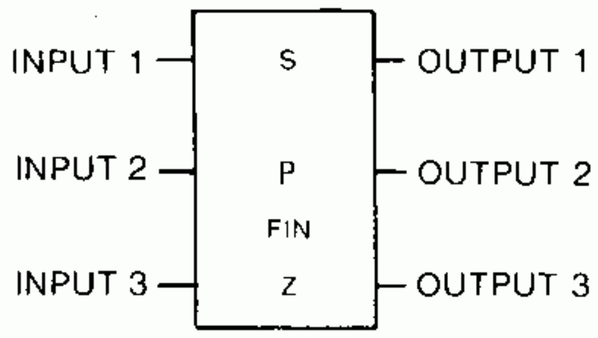
T → T shown in (a) will execute transfer of data shown in (b) input relay 10001 turns from OFF to ON. At this time, output 1 only will turn ON.

5.9.7 First In (FIN)

(1) Function

FIN is used in pair with a First Out (FOUT) (see Par. 5.9.8). This is like move of the R → T. The difference is that the data stored in the destination by FIN can be retrieved by FOUT in the order the data are stored.

(2) Form



S: Source table head reference number
P: Pointer
Z: Source table size

Fig. 5.55 FIN General Form

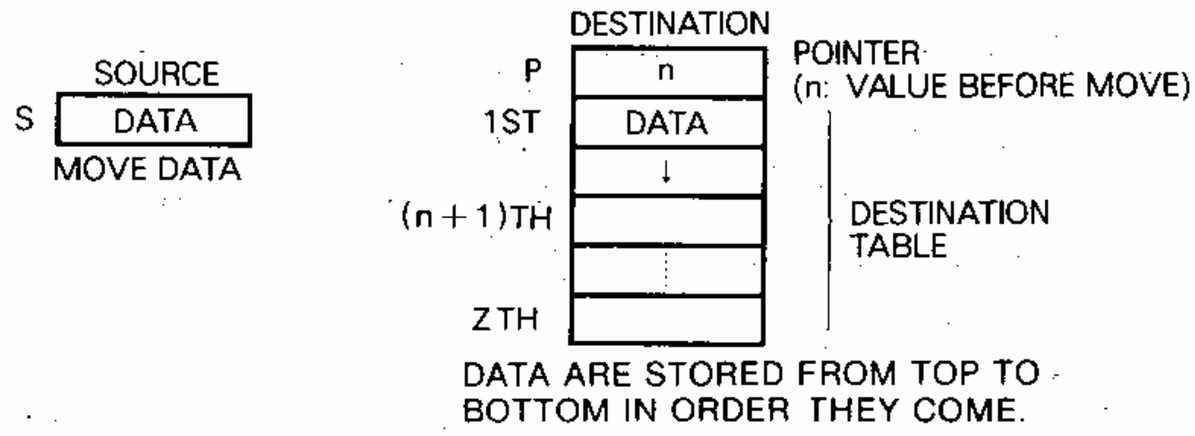
- Fig. 5.55 shows the form of first in.
- FIN is the symbol denoting the first in.
- FIN operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.67, specify the needed number for each element.

Table 5.67 Elements of FIN

Element	Description	Specified Number	
Top	Source reference number	• Coil	(00001-02033)
		• Input relay	(10001-10497)
		• Link coil	(D0001-D1009)
		• Input register	(30001-30128)
		• Holding register	(40001-42048)
		• Link register	(R0001-R1024)
Middle	Pointer	• Holding register	(40001-42047)
		• Link register	(R0001-R1023)
Bottom	Table size	• Constant	(1-100)

Note
1. When a relay reference No. is to be specified in the top stage, the lower-place 4 digits of the number that can be specified will be limited to [16 m + 1]. (m = 0, 1, 2, ...)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.
3. Destination starts with the reference No. next to the pointer.
4. Pointer is not included in the table size.

(3) Operation



Note Move will be performed at the rate of one register (or a set of 16 discretes) per scanning cycle.

(a) Inputs

When the input 1 is ON, the contents of the destination table are shifted down by one, according to the content of pointer, the nth data (the oldest one) to the (n+1)th register, the (n-1)th data to the nth register, and so on, until the first data are shifted down to the second register. Then the source data is moved to the first register emptied. Thus the destination table is managed in such a manner that the data are stored in the table from top to bottom in the order they come. The pointer is incremented by one after the shift and move of data. This process is performed all in one scanning cycle, regardless of the table size. If n ≧ table size, no data will be moved even when the input 1 is ON. Inputs 2 and 3 are not used.

(b) Outputs

FIN function uses all three outputs. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever the table is full (pointer equal to table length); the output 3 supplies power flow whenever the table is empty (pointer equals zero). The outputs 2 and 3 do NOT require any inputs to receive power flow, they only require appropriate pointer values.

Table 5.68 shows FIN operation.

Table 5.68 FIN Operation

Input 1	Operation	Condition	Output1	Output2	Output3
ON	Execution*	Pointer value=Table size	ON	ON	OFF
		Pointer value=0		OFF	ON
		Other than the above		OFF	OFF
OFF	–	Pointer value=Table size	OFF	ON	OFF
		Pointer value=0		OFF	ON
		Other than the above		OFF	OFF

* If pointer ≧ table size, this operation will not be executed.

(4) Example

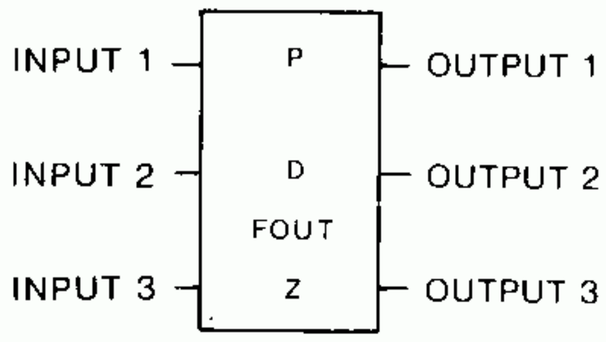
Example is shown in the section of FOUT.

5.9.8 First Out (FOUT)

(1) Function

This is an N-to-1 move that the destination table of FIN becomes the source table of FOUT.

(2) Form



P: Pointer
D: Destination table head reference number
Z: Source table size

Fig. 5.56 FOUT General Form

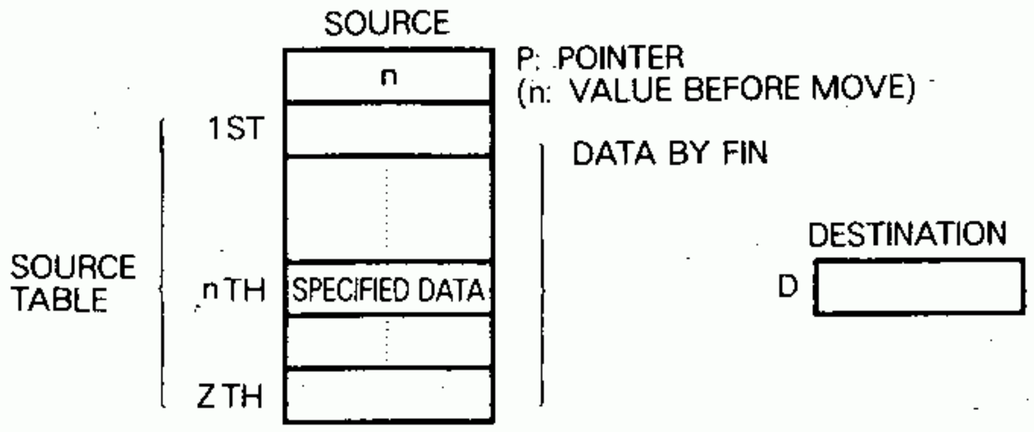
- Fig. 5.56 shows the form of first out.
- FOUT is the symbol denoting the first out.
- FOUT operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.69, specify the needed number for each element.

Table 5.69 FOUT Elements

Element	Description	Specified Number
Top	Pointer	• Holding register (40001-42047) • Link register (R0001-R1023)
Middle	Destination reference number	• Coil (00001-02033) • Link coil (D0001-D1009) • Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-100)

Note
1. When a relay reference No. is to be specified in the top stage, the lower-place 4 digits of the number that can be specified will be limited to {16 m + 1}. (m = 0, 1, 2, ...)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.
3. Source starts with the reference No. next to the pointer.
4. Pointer is not included in the table size.

(3) Operation



Note Move will be performed at the rate of one register per scanning cycle. The data stored in a FIFO table will be retrieved by FOUT with the oldest one first, i.e. by the First in-First out principle.

(a) Inputs

When the input 1 is ON, the nth (but not the (n+1)th) contents of the source table are moved to the destination where n is the value of the pointer indicating the number of data stored. The pointer is decremented by one after moved of the data. If n=0, data will not be moved even when the input 1 is ON. The inputs 2 and 3 are not used.

Note The nth source register, whose contents are retrieved by FOUT, holds 0 unless new data is placed by FIN.

(b) Outputs

FOUT function uses all three outputs: each of the three outputs behaves in the same way on FIN function block. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever the table is full (pointer equal to table length); the output 3 supplies power flow whenever the table is empty (pointer equals zero). The outputs 2 and 3 do NOT require any inputs to receive power flow, they only require appropriate pointer values.

Table 5.70 shows FOUT operation.

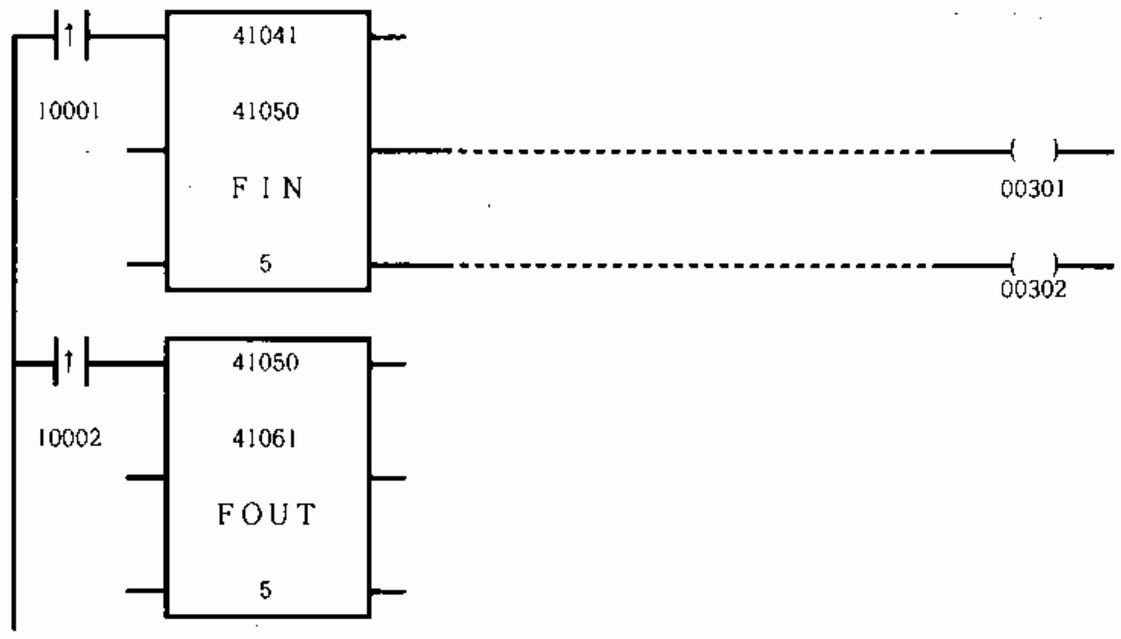
Table 5.70 FOUT Operation

Input 1	Operation	Condition	Output1	Output2	Output3
ON	Execution*	Pointer value=Table size	ON	ON	OFF
		Pointer value=0		OFF	ON
		Other than the ablove		OFF	OFF
OFF	–	Pointer value=Table size	OFF	ON	OFF
		Pointer value=0		OFF	ON
		Other than the above		OFF	OFF

*If pointer > table size, this operation will not be executed.

(4) Example (FIN, FOUT)

(a) Ladder



(b) Content of transfer (FIN)

① When the content of the FIFO table before execution of (the first) transfer is as follows.

	SOURCE
41041	100

	DESTINATION	
41050	0	POINTER
41051	0	DESTINATION TABLE
41052	0	
41053	0	
41054	0	
41055	0	

② When the first transfer is executed after turning 10001 from OFF to ON.

	SOURCE
41041	100

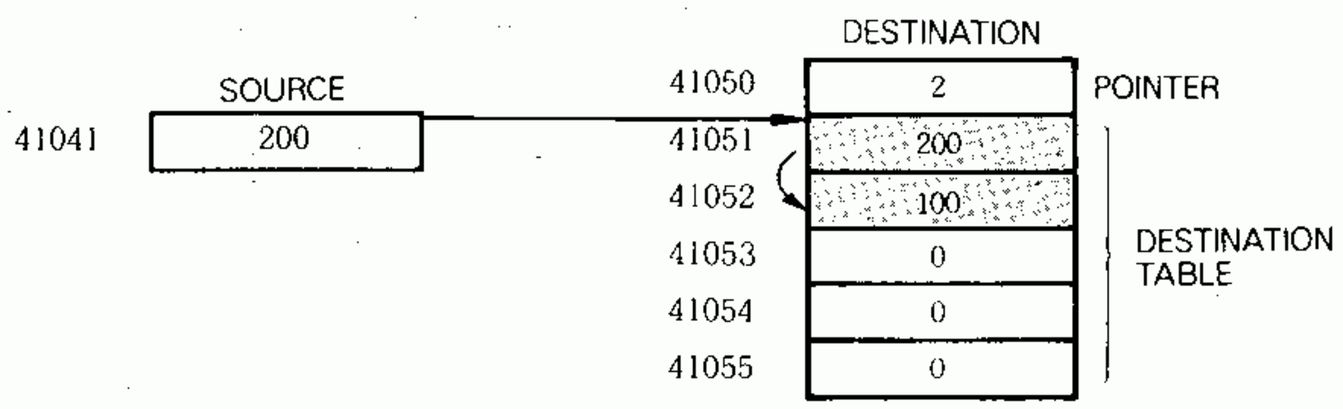
	DESTINATION	
41050	1	POINTER
41051	100	DESTINATION TABLE
41052	0	
41053	0	
41054	0	
41055	0	

③ The content of the register before the second transfer is shown below.

	SOURCE
41041	200

	DESTINATION	
41050	1	POINTER
41051	100	DESTINATION TABLE
41052	0	
41053	0	
41054	0	
41055	0	

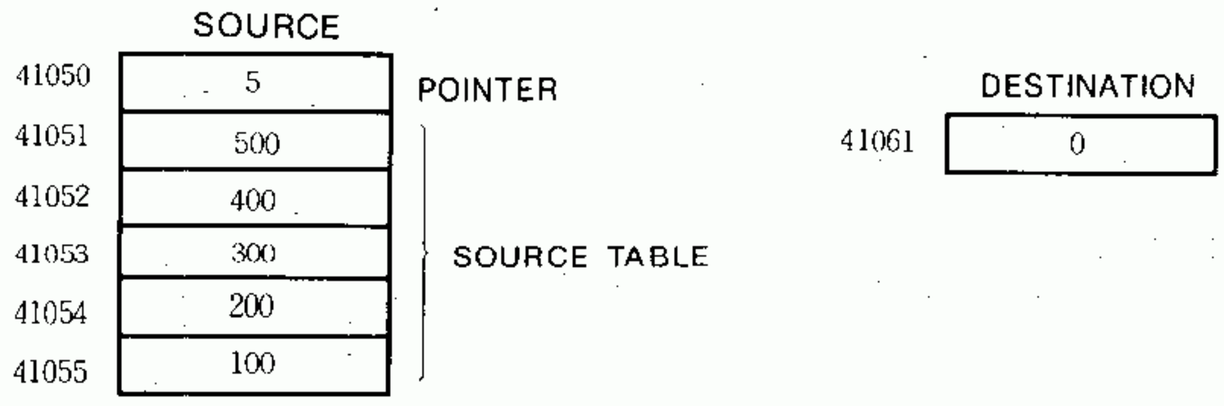
④ When the second transfer is executed after turning 10001 from OFF to ON, the data transferred first are shifted down and the present source data are stored in the head of the destination table.



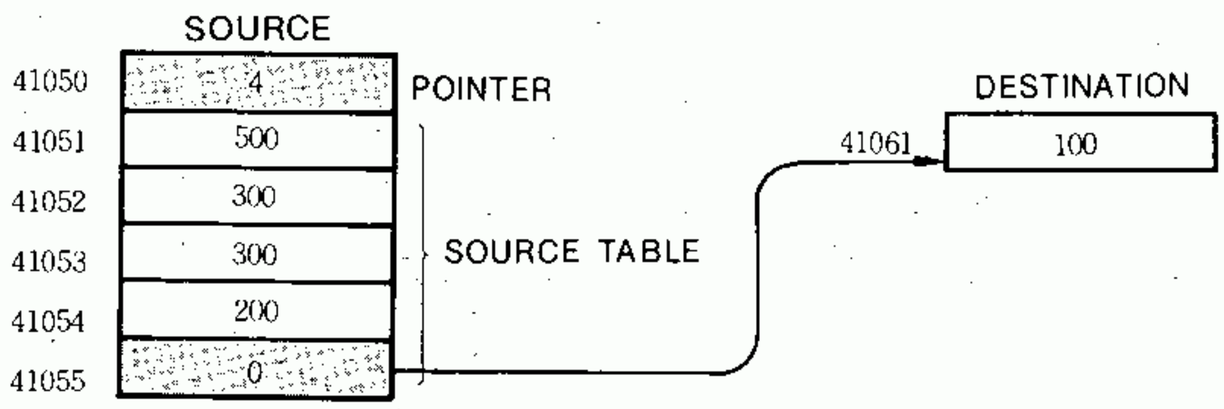
Coil 00301 turns ON when the value of pointer is 5, and coil 00302 turns ON when the value of pointer is 0.

(c) Content of tuanster (FOUT)

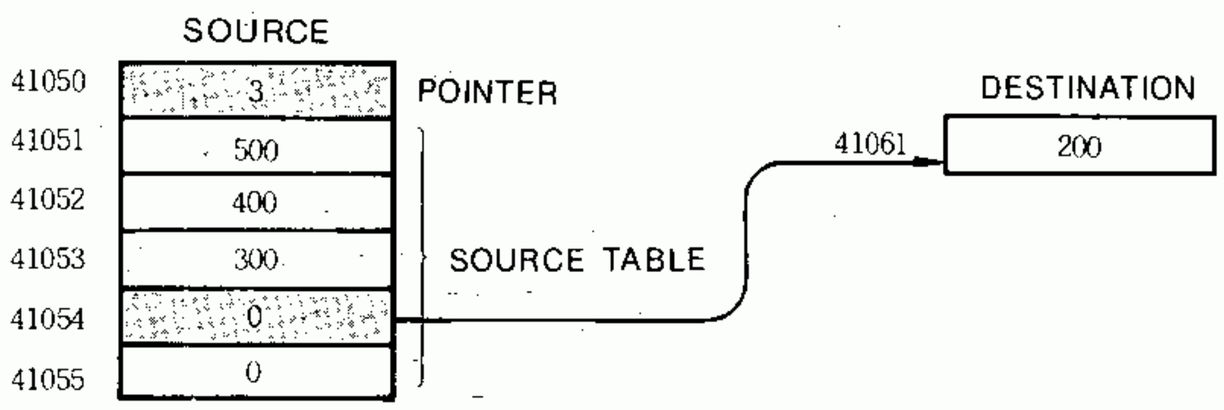
① When the content of the FIFO table before execution of (the first) transfer is as follows.



② When the first transfer is executed after turning 10002 from OFF to ON.



③ When the second transfer is executed after turning 10002 ON.



5.9.9 Table Search (SRCH)

(1) Function

This function searches a table of registers for a specified value. The source is not altered, only examined. If necessary, the SRCH function searches the entire table in one scan. It uses a pointer to indicate the location(s) within the table of registers which contain the value for which it is searching. This pointer is the only register whose value is altered by the SRCH function.

(2) Form

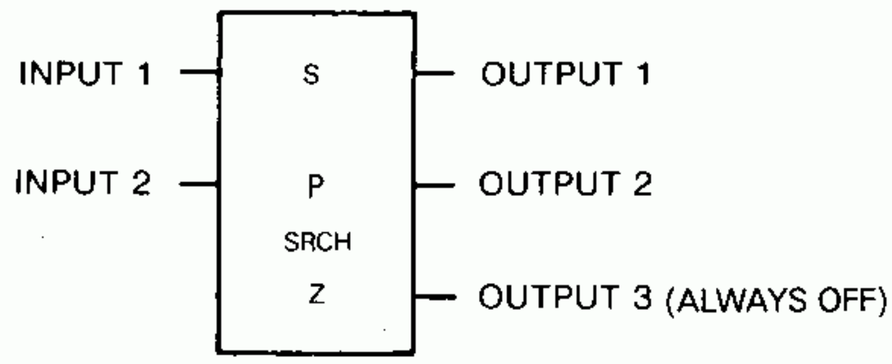


Fig. 5.57 SRCH General Form

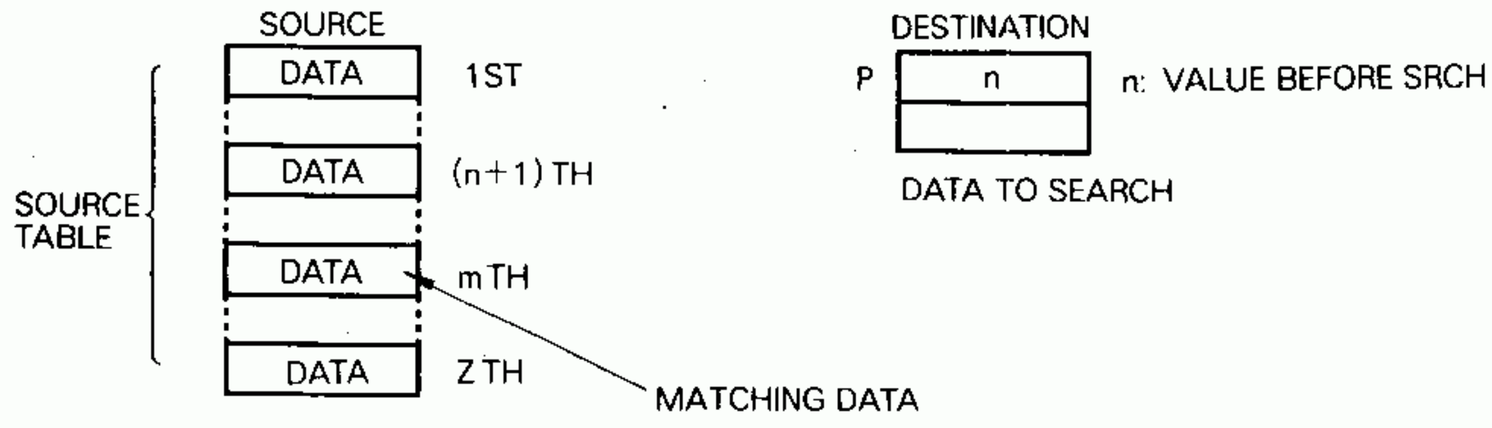
- Fig. 5.57 shows the form of search.
- SRCH is the symbol denoting the search.
- SRCH operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.71, specify the needed number for each element.

Table 5.71 Elements of SRCH

Element	Description	Specified Number
Top	Source table head reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Pointer	• Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-100)

Note Destination table starts with the reference No. next to the pointer.

(3) Operation



Before search, store the data to be searched in the location next to the destination pointer. Assume the value of the pointer is n when SRCH is required, then search is started from the (n + 1)th source data. When a matching data is found at the mth location, the search will be completed with m placed in the pointer. If a matching data is not found, the search will be completed with 0 placed in the pointer.

SRCH searches for one piece of matching data during every scanning cycle. If all data of the source table matches, it takes N scanning cycles (N is the table size) to complete the search. If there is no matching data, the search is completed in one scanning cycle.

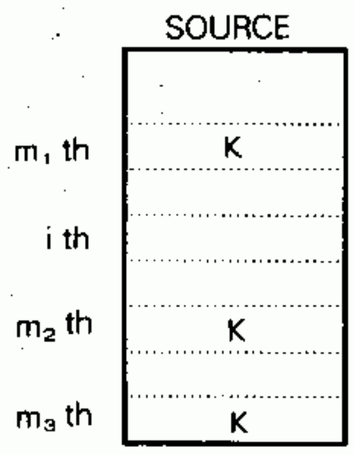
Where to start the next search depends on the status of inputs 1 and 2.

(a) Inputs

When only the input 1 is ON, n is set to 0 automatically before searching. SRCH always starts at the top of the source table. Even if there are many pieces of matching data in the source table, only the first one will be detected by search.

When the inputs 1 and 2 are ON, SRCH starts at the (n+1)th data of the source table when the pointer is n before searching. If the search is required from the ith data, set the pointer to i-1 and turn on the inputs 1 and 2.

If the m_1th, m_2th, and m_3th data of the source table are equal to the specified data K, set the pointer to i-1 and turn on the inputs 1 and 2. The search starts at the ith data and ends with $n = m_2$. The next search will start at the (m_2+1)th data and ends with $n = m_3$.



In any cases (only the input 1 is ON, or the inputs 1 and 2 are ON);

- If matching data is not found, the search of the scanning cycle is completed with n =N (N is the table size) then n is cleared to 0. (The search is not performed again from the top of the table during the same scanning cycle.)
- If the input is turned on when n ≧ N, n is automatically cleared to 0 before searching. As a result, the search is started at the top of source data. The input 3 is not used.

Note The pointer is 0 whenever the input 1 is OFF.

(b) Outputs

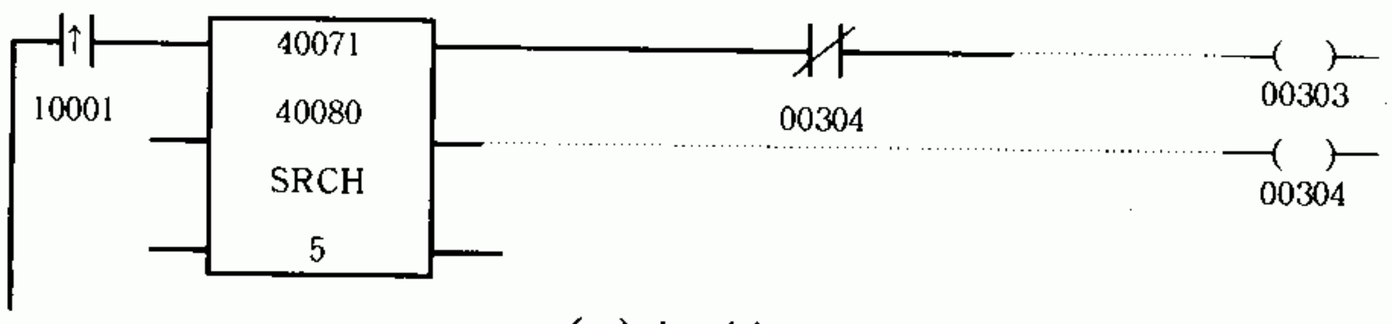
The SRCH function utilizes only the outputs 1 and 2. The output 3 has no significance and will be OFF (no power flow) under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever a match is found; if no match is found, this output will not supply power flow and the pointer will contain the number zero.

Table 5.72 shows SRCH operation.

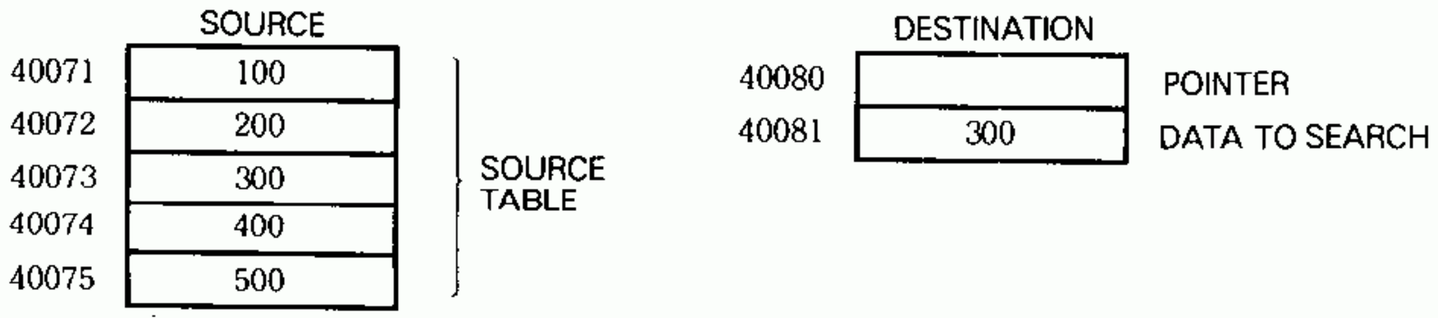
Table 5.72 SRCH Operation

Input 1	Input 2	Operations	Miscompare	Output 1	Output 2
ON	OFF	Search from the first source register.	Yes	ON	ON
			No		OFF
	ON	Search from the next source register to one indicated by the pointer.	Yes		ON
			No		OFF
OFF	—	Not executed.	—	OFF	OFF

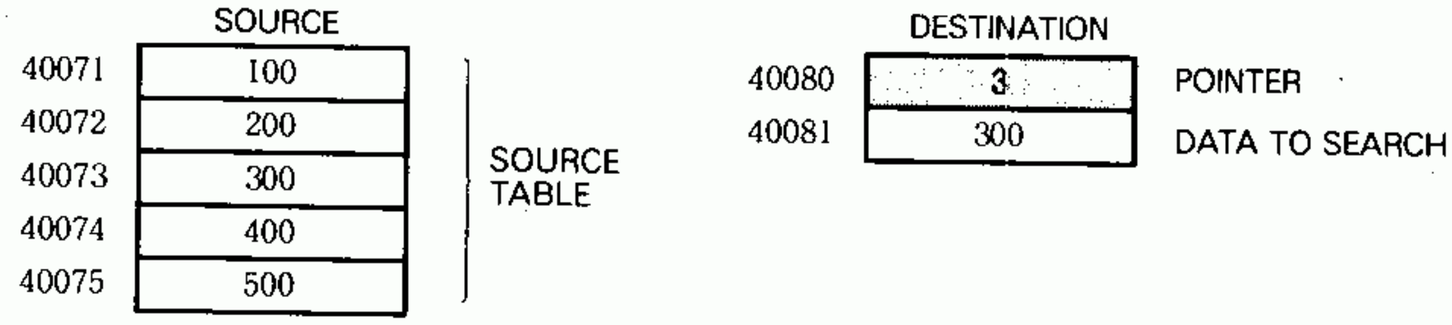
(4) Example



(a) Ladder



① Data before Execution

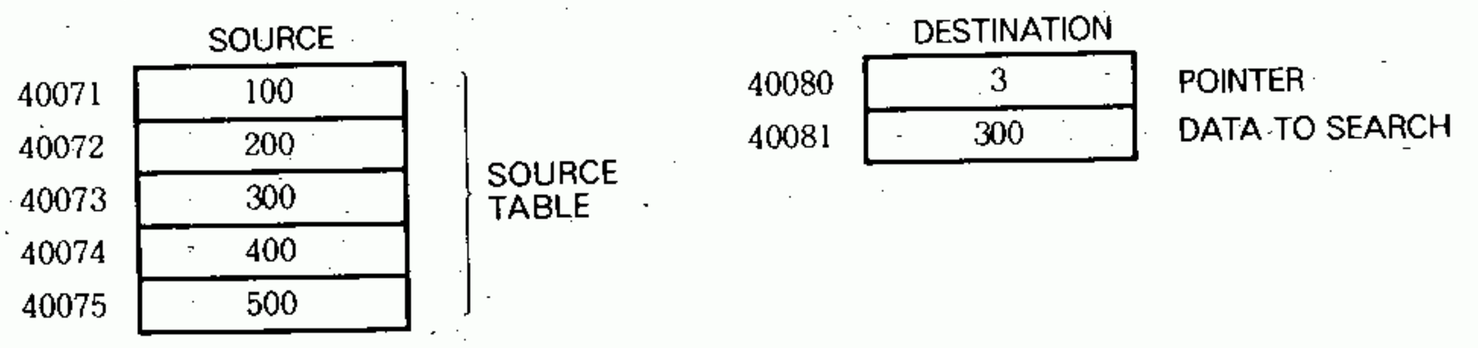


② Data after Execution

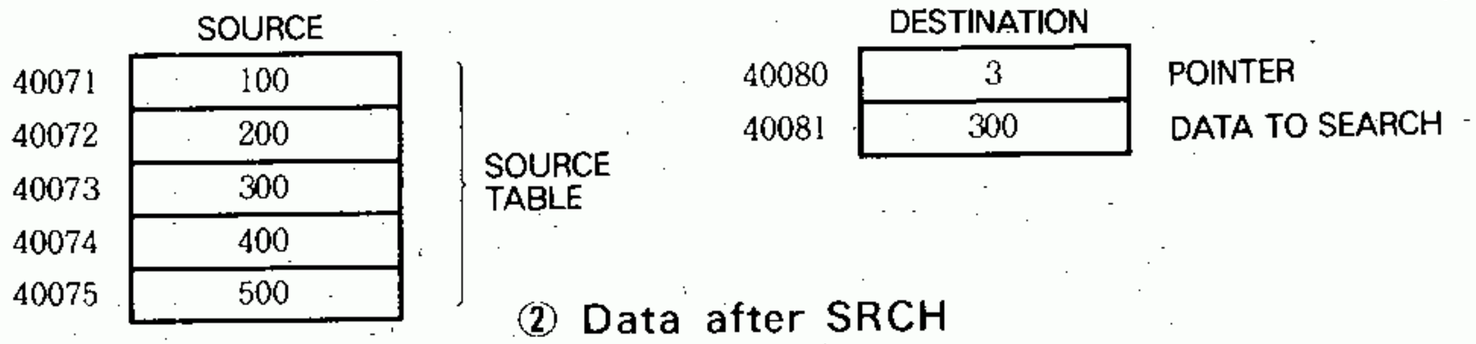
(b) SRCH Operation (with SRCH Data)

When input relay 10001 is turned ON, SRCH shown in (a) detects the data that coincide with the content of destination in the third register from the head of the source table as shown in (b) and sets 3 in the pointer. At this time, coil 00304 only turns ON.

5.9.9 Table Search (SRCH) (Cont'd)



① Data before Execution



② Data after SRCH

(b) SRCH Operation (without SRCH Data)

When input relay 10001 is turned from OFF to ON, SRCH shown in (b) starts search from the fourth register from the head of source table according to the instructions of the pointer. Nothing that coincides with the content of destination register in the fourth register onward of source table, so that the value of the pointer remains 3 without being updated. At this time, coil 00303 only turns ON. Coil 00303 can be used as Search End signal if there is no coinciding data.

5.9.10 Table Set (TSET)

(1) Function

Moves the source content in 1 scan cycle to all destination tables.

(2) Form

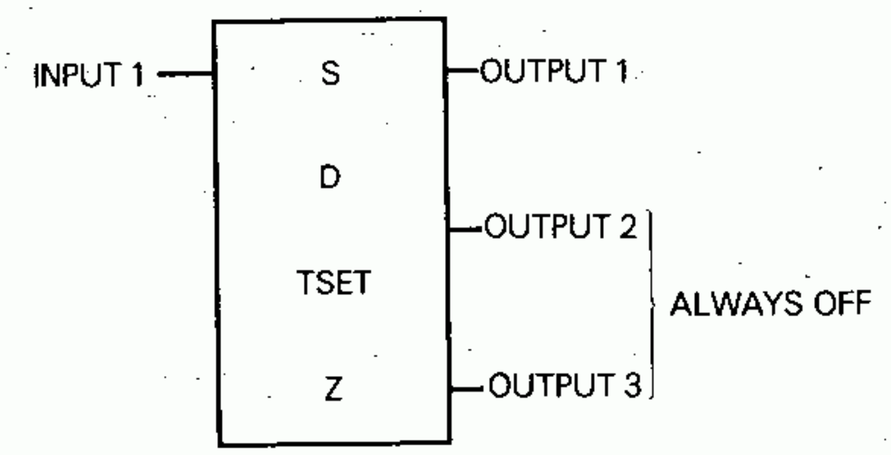


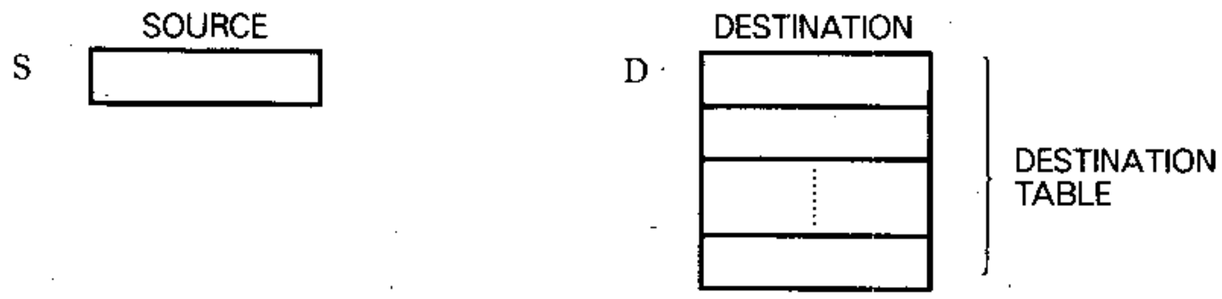
Fig. 5.58 TEST General Form

- Fig. 5.58 shows the form of table set.
- TSET is the symbol denoting the table set.
- TSET operation requires three elements placed vertically (top, middle and bottom). Referring to Table 5.73, specify the needed number for each element.

Table 5.73 Elements of TSET

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination table head reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

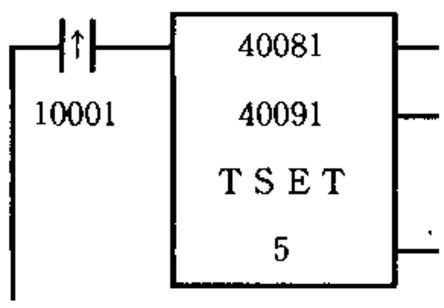
(3) Operation



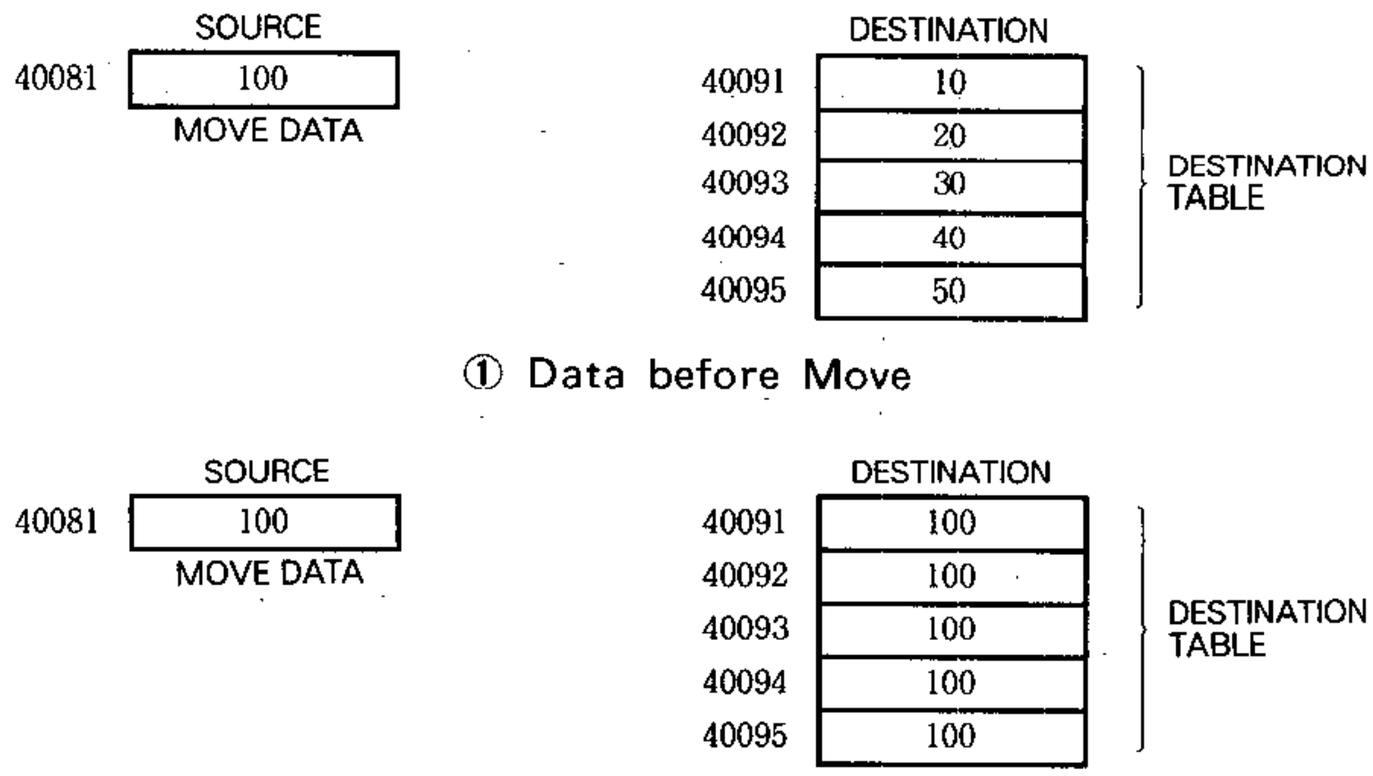
- When the input 1 is ON, TSET moves the source content to all destination tables.
- When the input 1 is ON, the output 1 is ON (Copy of the input 1.)

  The outputs 2 and 3 are always OFF.

(4) Example



(a) Ladder



① Data before Move

② Data after Move

(b) TSET Operation

If the input relay 10001 is turned to ON, TSET in (a) moves the source register content to all registers of destination table as shown in (b).

5.9.11 Get Controller System Status (STAT)

(1) Function

This function provides the user with access to the GL40S system status. The STAT is useful in the design of system diagnostics.

(2) Form

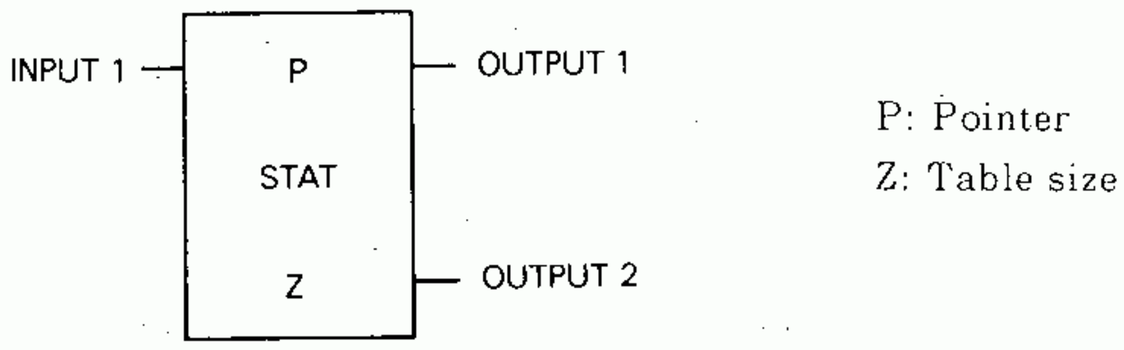


Fig. 5.59 STAT General Form

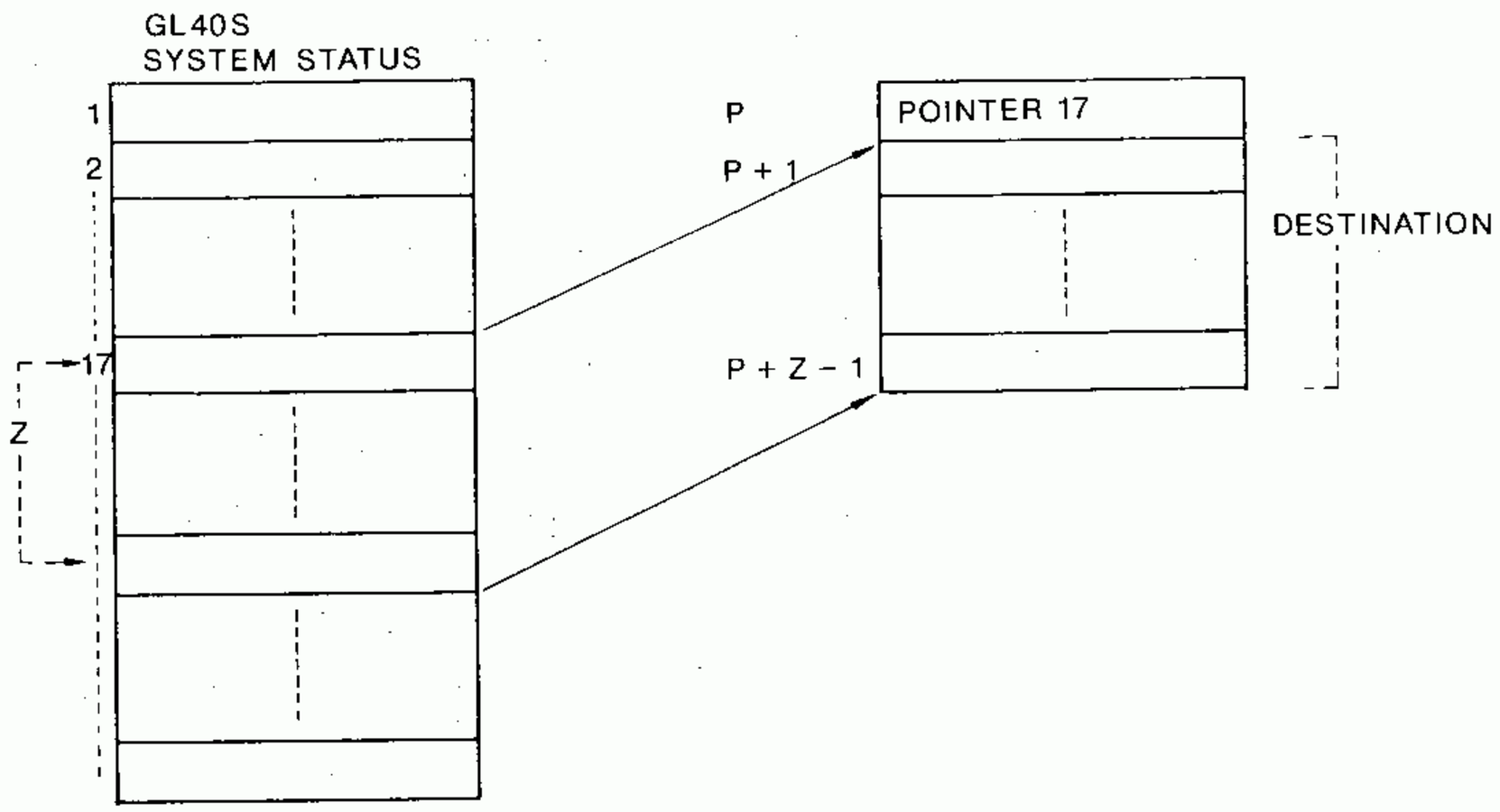
- Fig. 5.59 shows the form of system status.
- STAT is the symbol denoting the system status.
- STAT operation requires two elements placed vertically (top and bottom). Referring to Table 5.74, specify the needed number for each element.
- **P (Pointer) : Specify the start point of getting controller system status which is the source.**
- **Z (Table size) : Specify the data number of system status which will be copied.**

Table 5.74 STAT Elements

Element	Description	Specified Number
Top	Pointer	• Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size (Number)	• Constant (1-100)

(3) Operation

Specified status data are moved to the destination in one scan.



(a) Inputs

Of the two available inputs, only the input 1 affects the operation of the STAT function. When this input node receives power flow, all statuses are moved to the destination in one scan.

(b) Outputs

When the input 1 is ON and the operation is executed, the output 1 is ON.
When the input 1 is ON and the operation isn't executed, the output 2 is ON.

Table 5.75 STAT Operation

Input 1	Operation	Condition	Output 1	Output 2
ON	Execution	Pointer value: 1-100 Table size: 1-100 2 ≦ Pointer value + Table size ≦ 101	ON	OFF
	No Execution	Other than the above	OFF	ON
OFF	No Execution	—	OFF	OFF

Note If the number of destination registers is less than the table size, operation is not executed.

(c) System Status

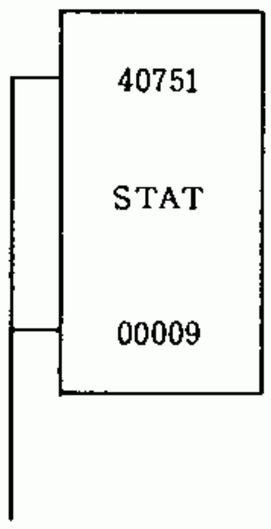
Table 5.76 shows the GL40S system status

Table 5.76 GL40S System Status

1	GL40S MACHINE STATUS	6390 (HEXADECIMAL)
2	FOR FUTURE EXPANSION *	6391
3	COMM STATUS	6392
4	PC-LINK STATUS	6393
5	LOCAL I/O STATUS *	6394
6	SERVO I/F 1 STATUS	6395
7	SERVO I/F 2 STATUS	6396
8	SERVO I/F 3 STATUS	6397
9	SERVO I/F 4 STATUS	6398
10	SCAN TIME	6399
11	FOR FUTURE EXPANSION *	639A
12	FOR FUTURE EXPANSION *	639B
13	FOR FUTURE EXPANSION *	639C
14	FOR FUTURE EXPANSION *	639D
15	FOR FUTURE EXPANSION *	639E
16	FOR FUTURE EXPANSION *	639F
17	LOCAL I/O SLOT ERROR STATUS (RACK 1)	63A0
18	LOCAL I/O SLOT ERROR STATUS (RACK 2)	63A1
19	LOCAL I/O SLOT ERROR STATUS (RACK 3)	63A2
20	LOCAL I/O SLOT ERROR STATUS (RACK 4)	63A3
21	FOR FUTURE EXPANSION *	63A4
22	LOCAL I/O RACK ERROR STATUS	63A5
23	LOCAL I/O RACK ACK STATUS	63A6
24	LOCAL I/O RACK ACK STATUS HISTORY	63A7
25	FOR FUTURE EXPANSION *	63A8
26	FOR FUTURE EXPANSION *	63A9
27	FOR FUTURE EXPANSION *	63AA
28	FOR FUTURE EXPANSION *	63AB
29	FOR FUTURE EXPANSION *	63AC
30	FOR FUTURE EXPANSION *	63AD
31	FOR FUTURE EXPANSION *	63AE
32	FOR FUTURE EXPANSION *	63AF
33	PC-LINK STATION STATUS	63B0
34	PC-LINK STATION STATUS	63B1
⋮	FOR FUTURE EXPANSION *	⋮
⋮	⋮	⋮
	FOR FUTURE EXPANSION *	63F3

* The value is "0".

(4) Example



40751	1 (POINTER)
40752	GL40S MACHINE STATUS
40753	FOR FUTURE EXPANSION
40754	COMM STATUS
40755	PC-LINK STATUS
40756	LOCAL I/O STATUS
40757	SERVO I/F 1 STATUS
40758	SERVO I/F 2 STATUS
40759	SERVO I/F 3 STATUS
40760	SERVO I/F 4 STATUS

Status information can be obtained constantly because the input 1 is connected to the power rail at the left.
Because the content of 40751 is 1, 9 status data from "GL40S MACHINE STATUS " to "SERVO I/F 4 STATUS" are picked out.

5.9.12 Programming Data Move Circuit and Precautions

(1) Inputs to the data move circuit may be outputs of relays, timers, counters, arithmetic operations, matrices, and other data move circuits.
(2) Coils need not be connected to three output nodes (1, 2 and 3) of a data move circuit. It is permitted to connect a relay contact to the output nodes at right or connect the output node directly to an input node of an arithmetic circuit, except relays.
(3) To execute an operation constantly, connect the input directly to the power rail at left. To execute it only during one scanning cycle, use a transitional contact as an input.
(4) The range of the source or destination table specified by a table size must be whithin the range of the reference numbers of input relays, coils, or registers.
(5) Since the output 1 supplies power flow whenever the input 1 receives power flow, cascaded operation is possible. It is possible to OR the outputs by connecting a vertical shunt element.
(6) When coil 0×××× is to be specified as destination, the number of the coil group cannot be the same as the number of a coil group used in another circuit. If it is attempted, the message "COIL IS USED" is displayed on the screen of the programming panel and input operation will be rejected.
(7) When coil 0×××× is specified as destination, the coil is turned on and off by data move function even if it is disabled.
(8) When a pointer is used, its value does not exceed the table size normally. If the value is made greater than the table size forcedly by an another circuit, move is not executed (SRCH executes it after resetting the pointer to 0) and the pointer becomes equal to the table size.
(9) The source remains unchanged data after the move.

5.9.13 Example–Application Circuits of Data Move

(1) Setting and Reading Data (Applications of R → T and T → R)

This is an example to set and read a time of externally-preset timers. Fig. 5.60 shows a sample layout of switches and indicator. (Data conversion circuits of BCD, BIN are omitted.)

Set a timer number and time by the digital switches and push the SET pushbutton switch to store the set time in the register associated with the timer number and indicate the time.

Reading:

Set a timer number and push the READ pushbutton switch to read out the set time corresponding to the timer number from a table and indicate the time. It is assumed that the timer number will be N and the set time will be TN.

(a) Hardware Configuration

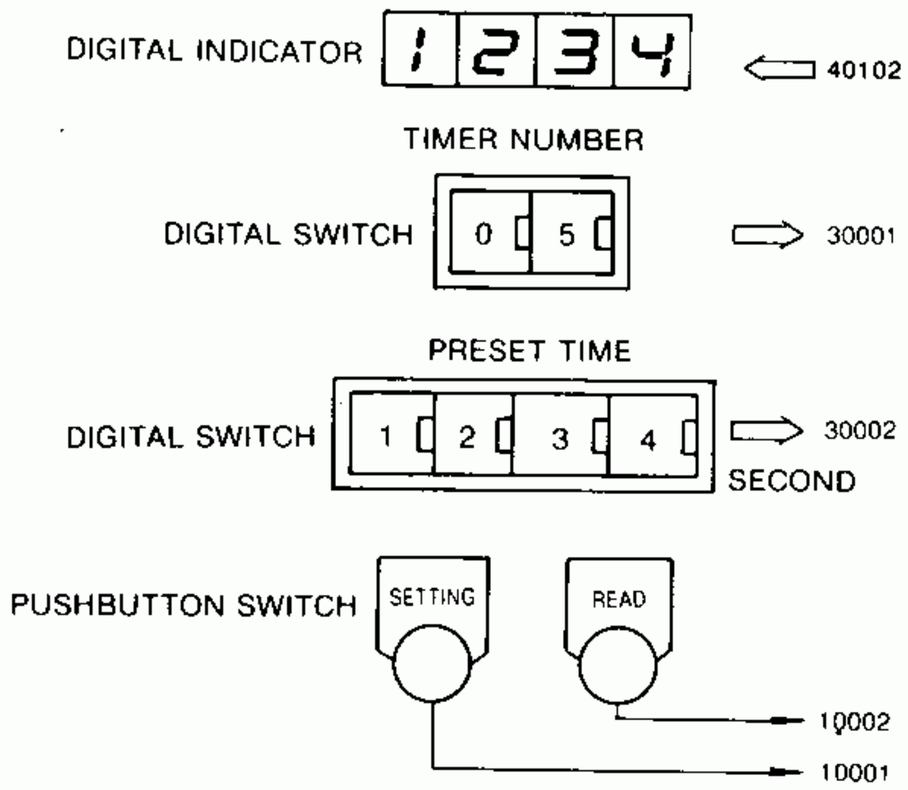
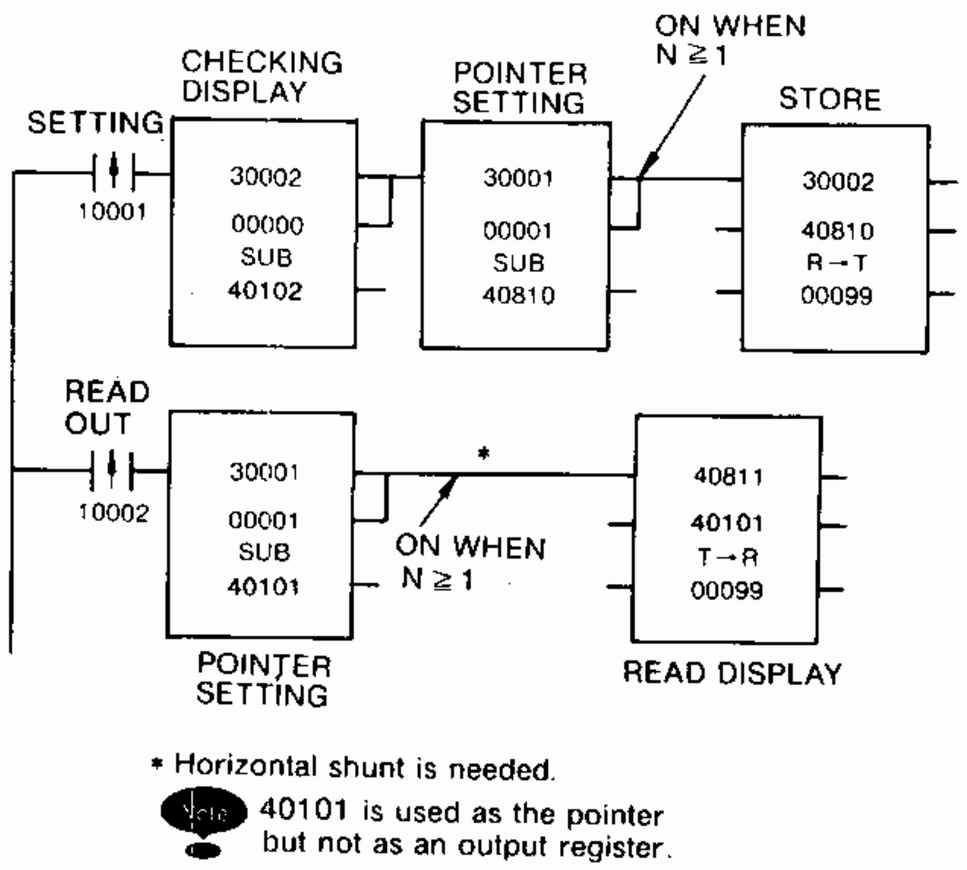


Fig. 5.60 External Devices and Assignments

(b) Ladder Diagram



* Horizontal shunt is needed.

Note 40101 is used as the pointer but not as an output register.

Fig. 5.61 Setting/Reading Circuits

(c) Data Flow

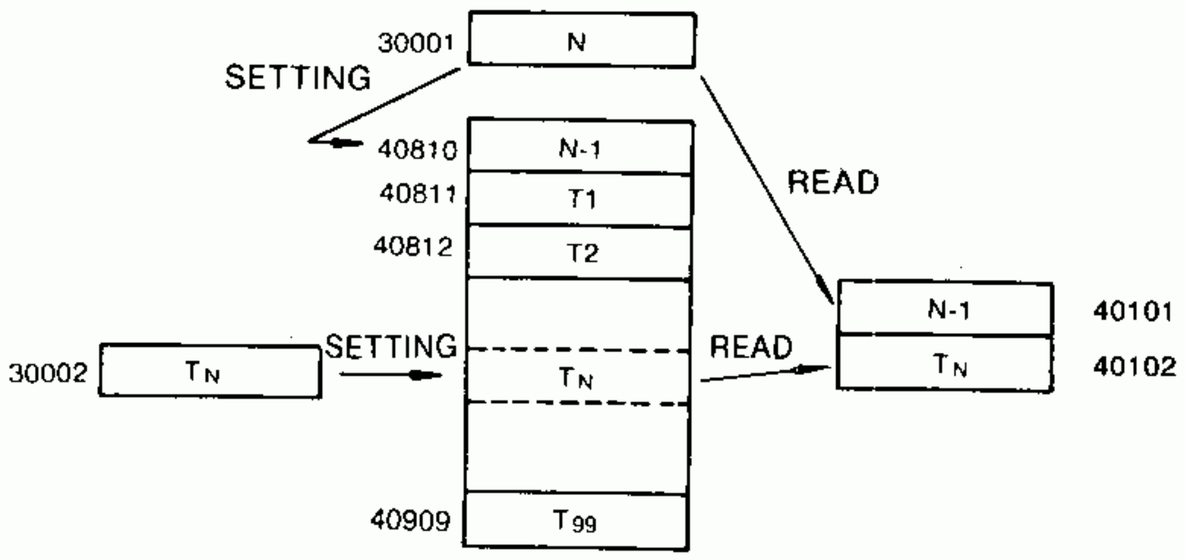


Fig. 5.62 Data Flow

(2) Move of Bit Pattern (Application of BLKM)

It is possible to execute a step-by-step sequence by storing the sequence of ON/OFF steps as a bit pattern in advance and taking out the bit pattern when executing. The example below is of three steps and with 48 output devices.

(a) Bit Pattern

Step \ Device No	48	47		3	2	1
1	ON	ON		OFF	ON	ON
2	ON	OFF		OFF	ON	OFF
3	OFF	OFF		ON	ON	OFF

MSB LSB

40911	----------011	FOR STEP 1
40912	----------	
40913	11----------	
40914	----------010	FOR STEP 2
40915	----------	
40916	10----------	
40917	----------110	FOR STEP 3
40918	----------	
40919	00----------	

BIT PATTERN TABLE
(0: OFF, 1: ON)

(b) Ladder Diagram

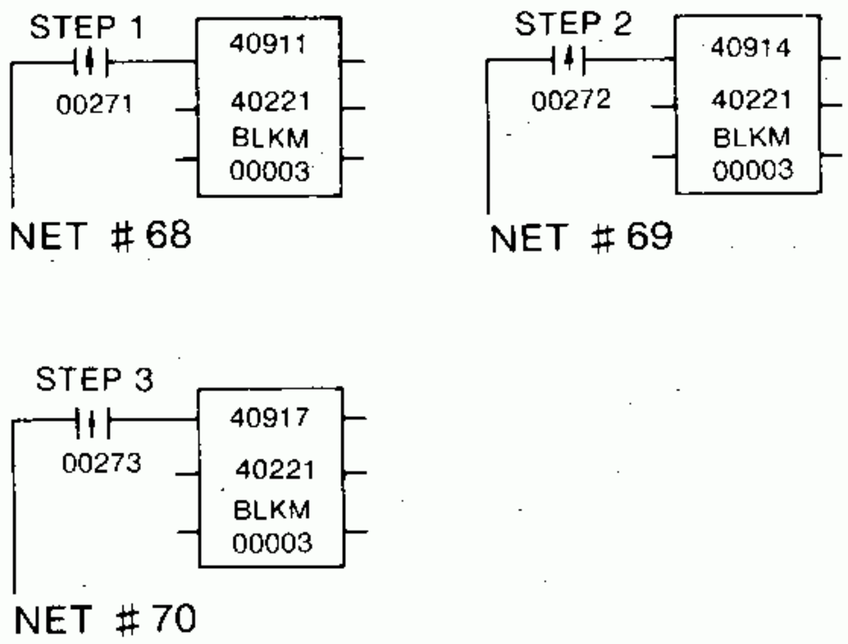
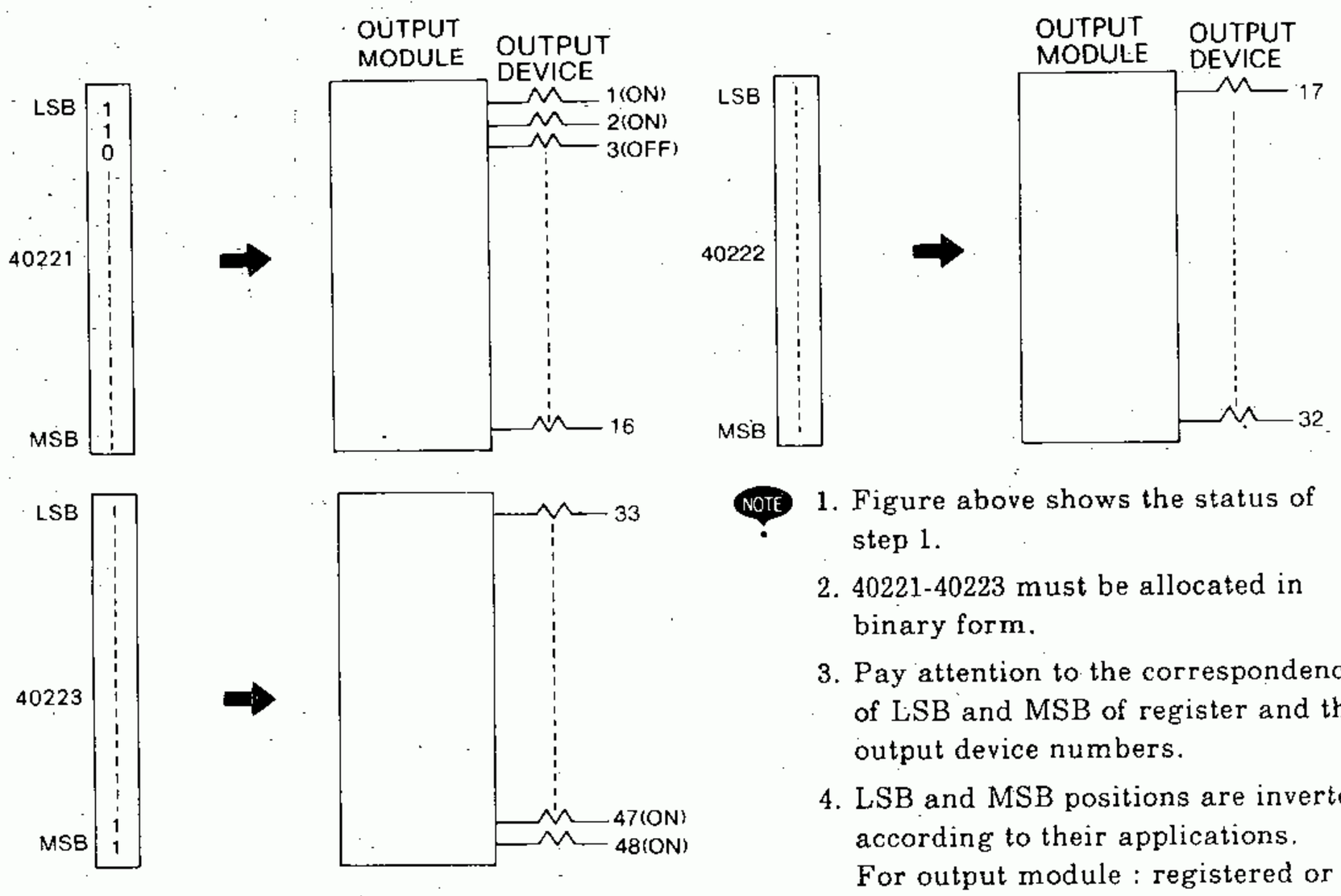


Fig. 5.63 Bit Pattern and the Ladder Diagram

(c) Output of Bit Pattern

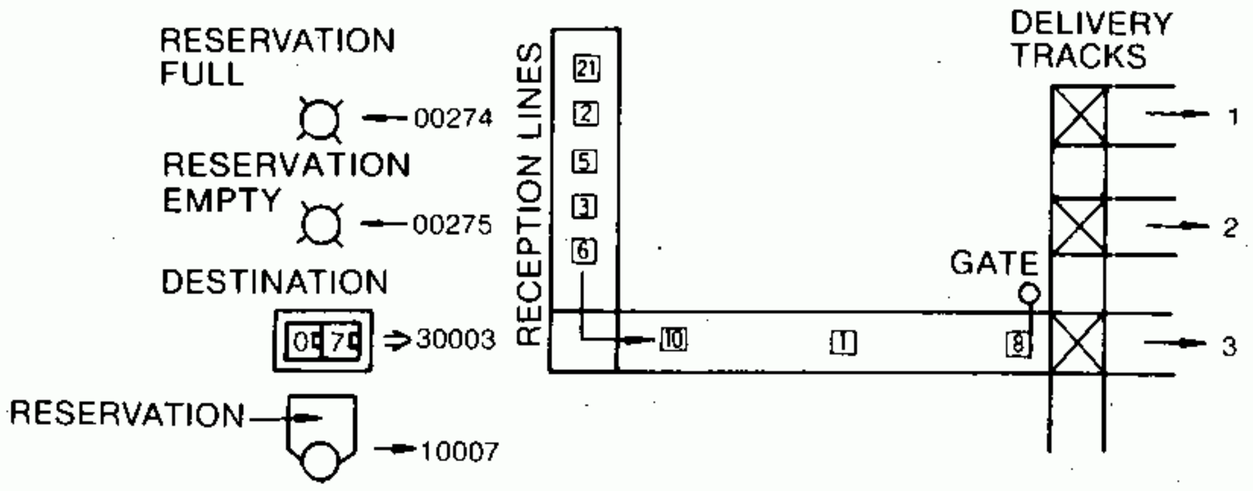


NOTE
1. Figure above shows the status of step 1.
2. 40221-40223 must be allocated in binary form.
3. Pay attention to the correspondence of LSB and MSB of register and the output device numbers.
4. LSB and MSB positions are inverted according to their applications.
   For output module : registered or discrete function.

Fig. 5.64 Bit Pattern Output

(3) Reservation (Applications of FIN and FOUT)

In this example, articles are grouped by destination and placed on delivery tracks. It is assumed that branching and interruption do not occur during transportation.



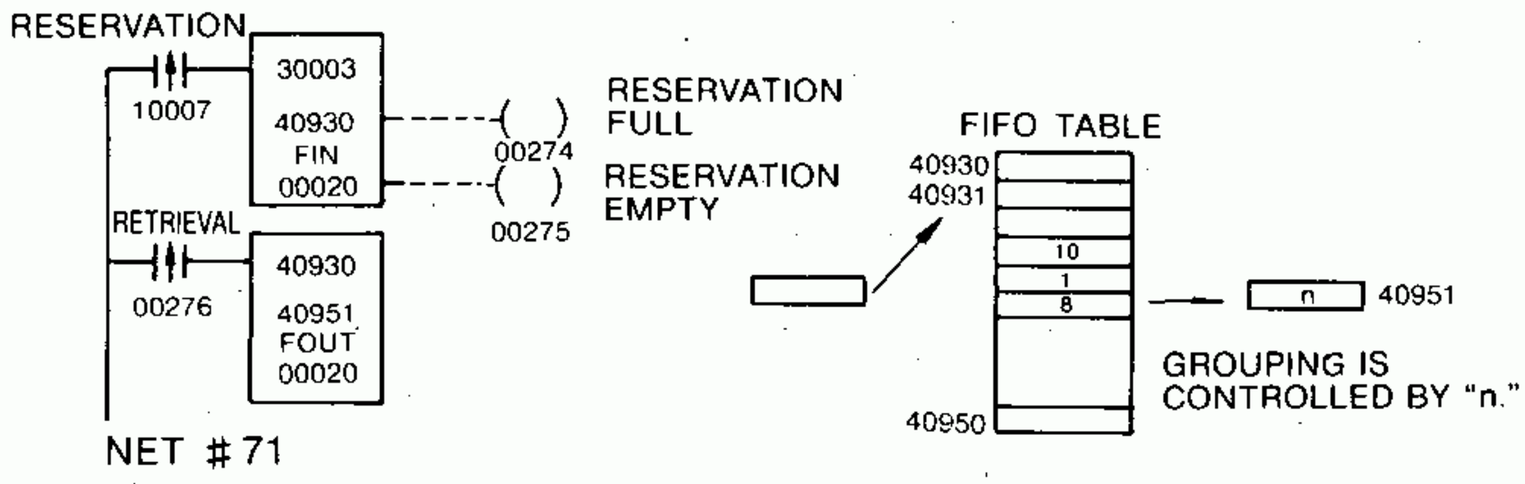


Fig. 5.65 Reservation Circuit

(4) Prevention of Double Reservation (Application of SRCH)

This is an example to prevent double reservation in the system shown in (3) above.

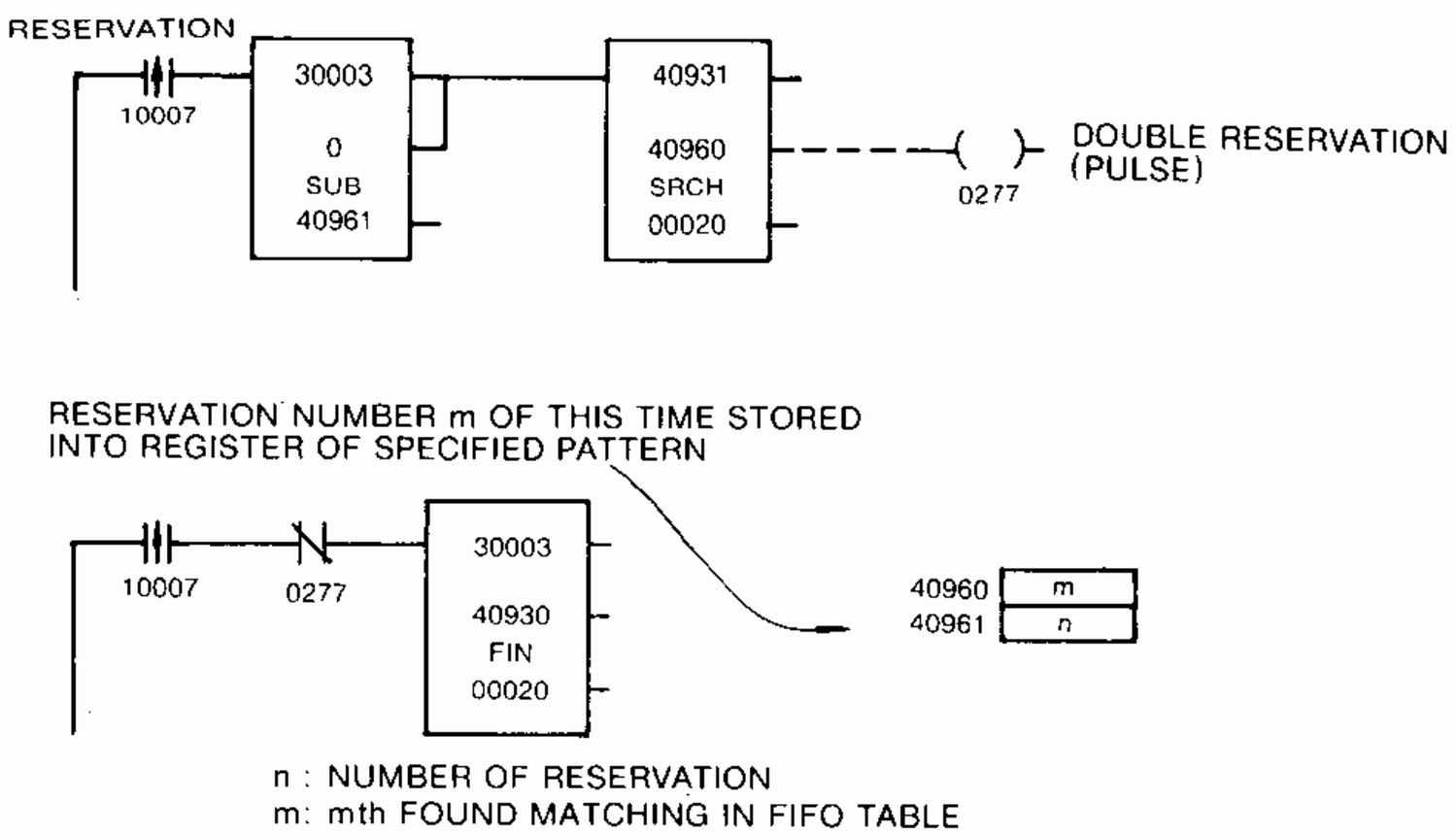


Fig. 5.66 Prevention Circuit of Double Reservation

5.10 INDEXED BLOCK MOVE

Indexed block move is a table-to-table move executed in one scanning cycle, like block move (BLKM) described Par. 5.9.3. In addition, there are following functions:

- The source or destination table can be specified according to the value (index) of the pointer.
- A group of input relays can be specified as the destination table.

5.10.1 Types of Indexed Block Move

Four types of the indexed block move are available as shown in Table 5.77.

Table 5.77 Types of Indexed Block Move

Type	Symbol	Description	Page
Block Move 1 with Destination Index	DIBT	Permits to specify destination table (input relay groups) according to pointer.	166
Block Move 2 with Destination Index	DIBR	Permits to specify destination table (hodling register groups) according to pointer.	171
Block Move 1 with Source Index	SIBT	Permits to specify source table (coils, input relays and input register groups) according to pointer.	174
Block Move 2 with Source Index	SIBR	Permits to specify source table (holding register groups) according to pointer.	178

5.10.2 Block Move 1 with Destination Index (DIBT)

(1) Function

This function can transfer all data of source table to the destination table specified by the pointer.

(2) Form

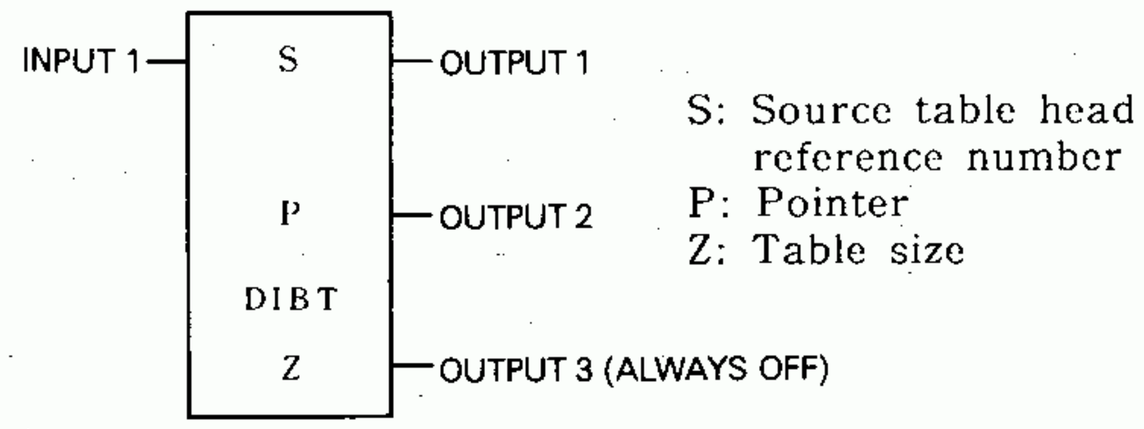


Fig. 5.67 DIBT General Form

- Fig. 5.67 shows the form of the block move 1 with destination index (DIBT).
- DIBT is the symbol denoting the block move 1 with destination index.
- DIBT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.78, specify any of constant K, reference numbers of discrete group and reference numbers of various registers for each of the elements.

Table 5.78 DIBT Elements

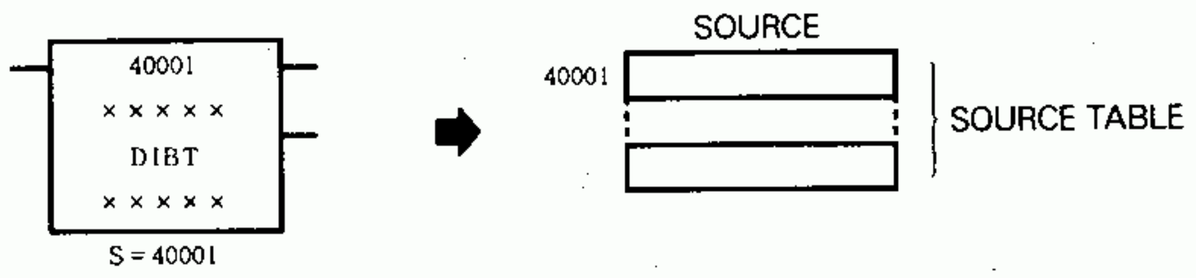
Element	Description	Specified Number
Top	Source table head reference number	Any of the following : • Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	With pointer, specified head number of destination table	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Size of source table and destination table	• Where input relay, constant (1-32) • Where link coil, constant (1-64) • Others (1-100)

1. When a coil and relay reference No. is to be specified in the top stage, the lower-place 4 digits of the number that can be specified will be limited to [16 m + 1]. (m = 0, 1, 2, ···)
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.

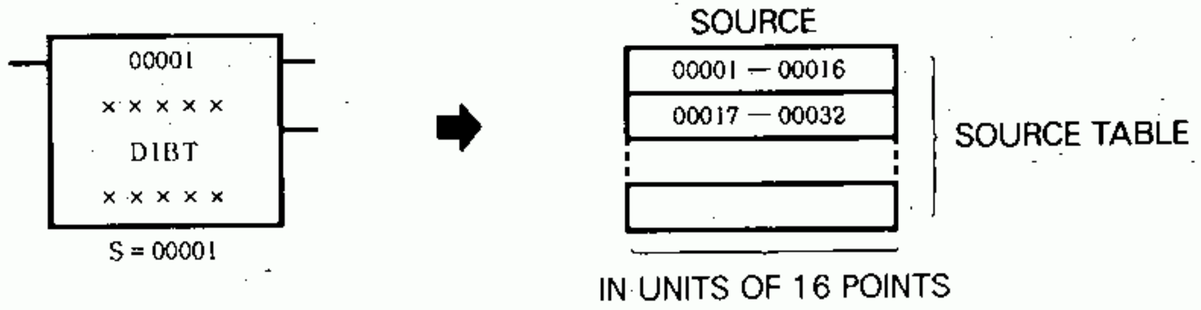
(3) Operation

The relation between source table and its head number is as follows:

Example 1:



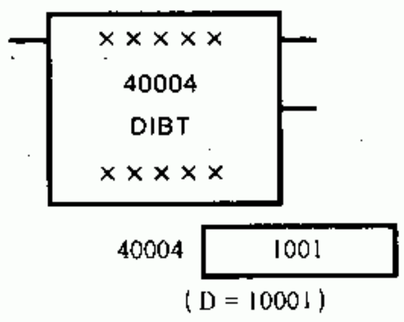
Example 2:

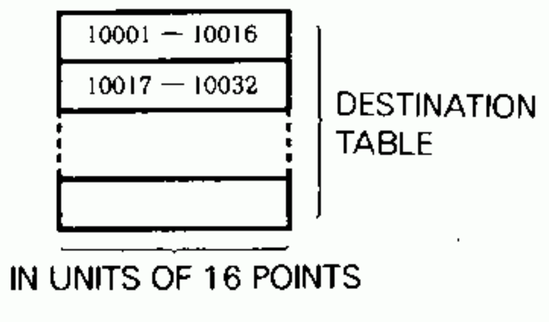


If a relay group or a coil group is to be specified, n = 16m + 1 (m = 0, 1, 2 ...) must hold, where n represents the lower-place 4 digits. The relation between pointer content (i) and the head number (D) of destination table (input relay group) is shown below.

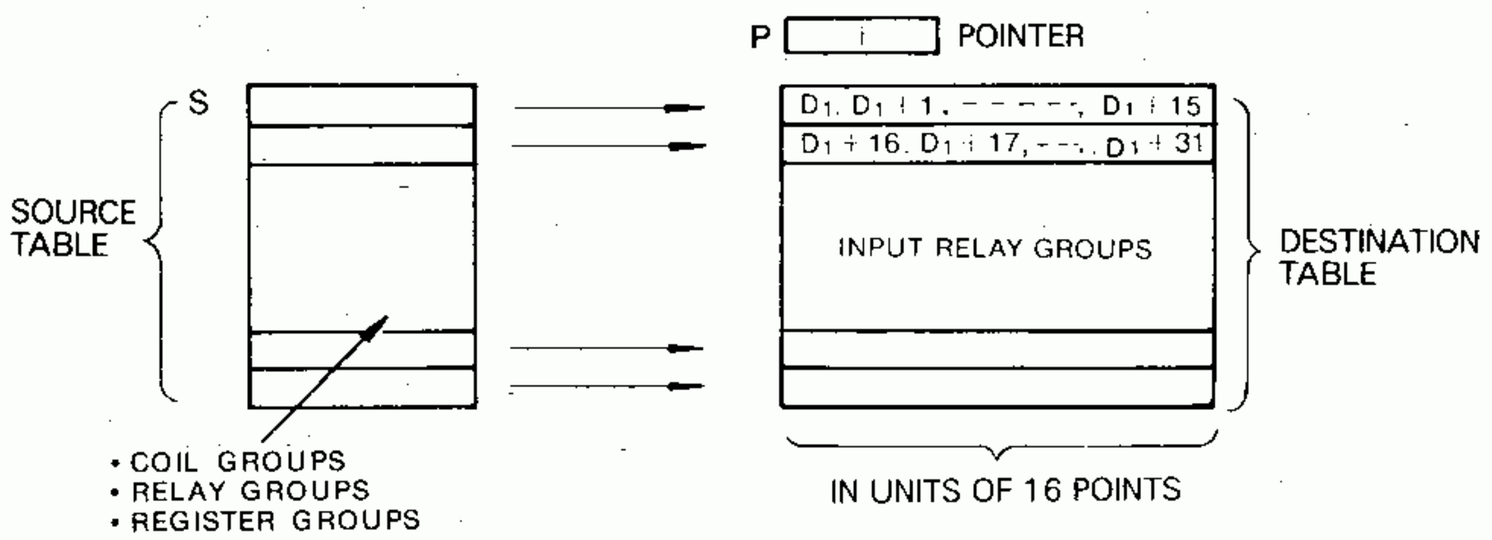
Example:

i	D
1001	10001
1002	10017
⋮	⋮
1032	10497





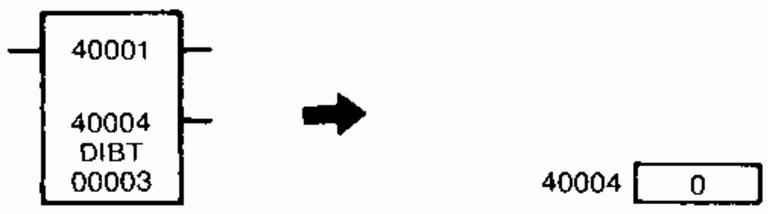
- By DIBT, all data of the source table will be moved to the destination table (input relay groups) specified with the value i of the pointer when the input 1 is ON. The output 1 is turned on. The move will be completed in one scanning cycle.



- In the following cases, the move will not be executed and the output 2 is turned on.

(a) When the value i of the pointer is out of the range of 1001-1032:

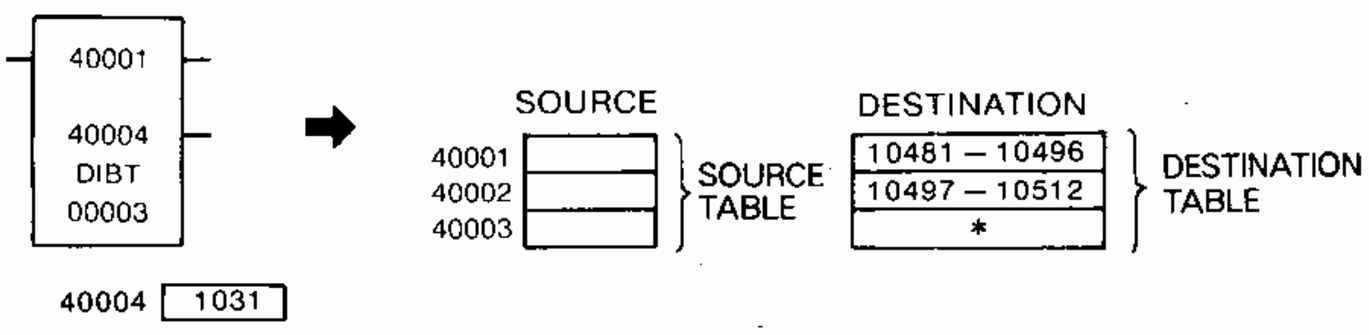
Example:



Note There is no input relay group corresponding to i=0. Therefore, no destination tables corresponding to the source tables 40001-40003 exist.

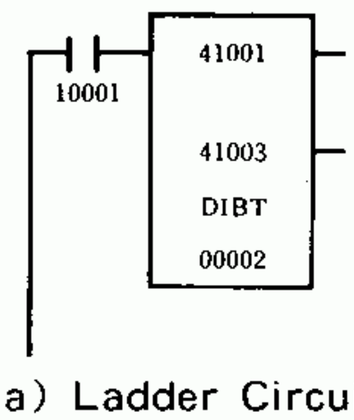
(b) When the value i of the pointer is within the range of 1001-1512 but no destination table to be specified by the value exists:

Example:

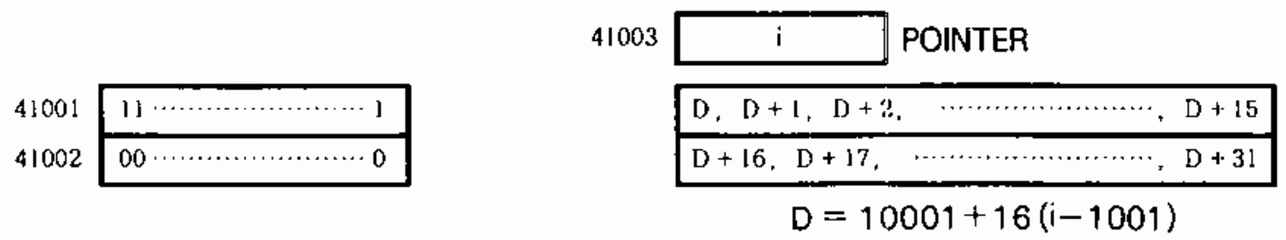


* No input ralay groups of destination corresponding to the source 40003 exist.

(4) Example



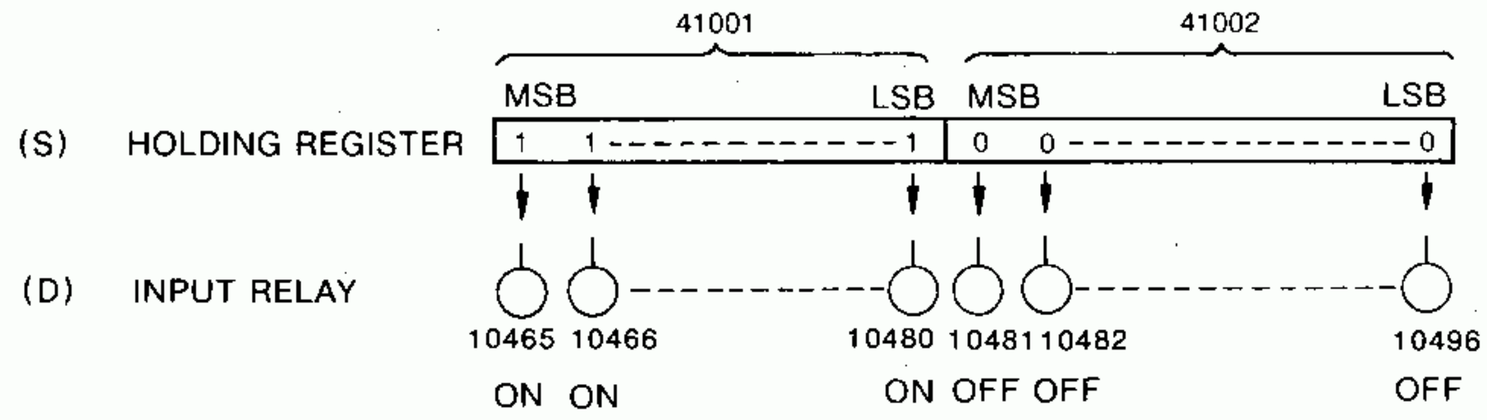
(a) Ladder Circuit



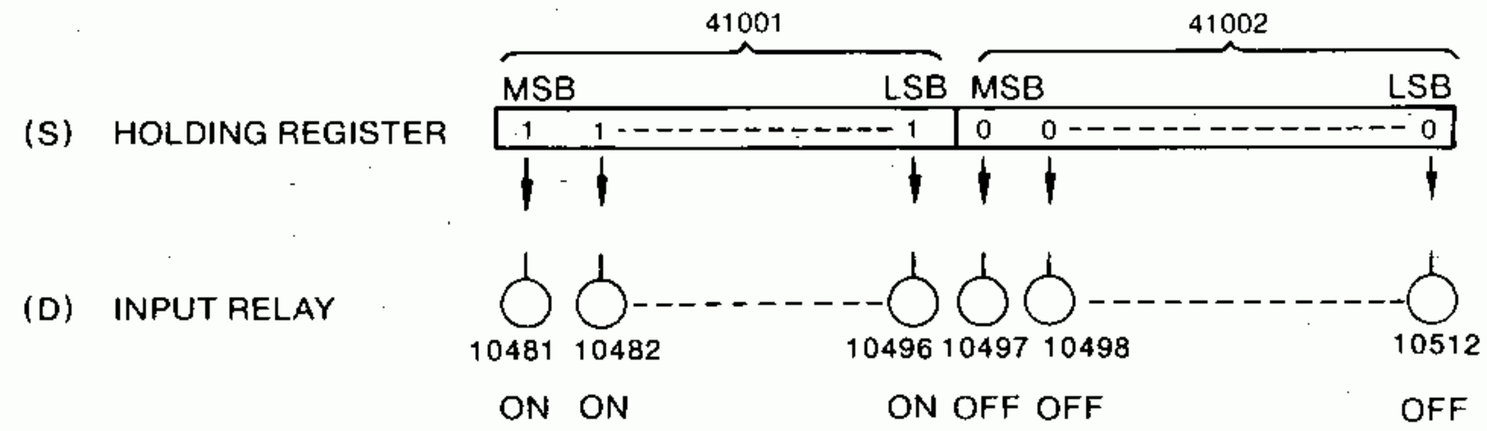
(b) Move Contents

5.10.2 Block Move 1 with Destination Index (DIBT) (Cont'd)

(a) When i = 1030, D1 = 10465. If the input relay 10001 is turned on at this time, the bit patterns of the holding registers 41001 and 41002 are moved to the input relay groups 10465-10496 and the output 1 is turned on.



(b) When i = 1031, D1 = 10481. If the input relay 10001 is turned on at this time, the bit patterns of the holding registers 41001 and 41002 are moved to the input relay groups 10481-10512 and the output 1 is turned on.



(c) When i = 1032, D1 = 10497. Because the table size is 2, no input relay groups of destination corresponding to the holding register 41002 exist. Therefore, even if the input realy 10001 is turned on, the move will not be performed and the output 2 is turned on.

5.10.3 Block Move 2 with Destination Index (DIBR)

(1) Function

This function can transfer all data of source table to the destination table (a group of holding regiter) specified by the pointer.

(2) Form

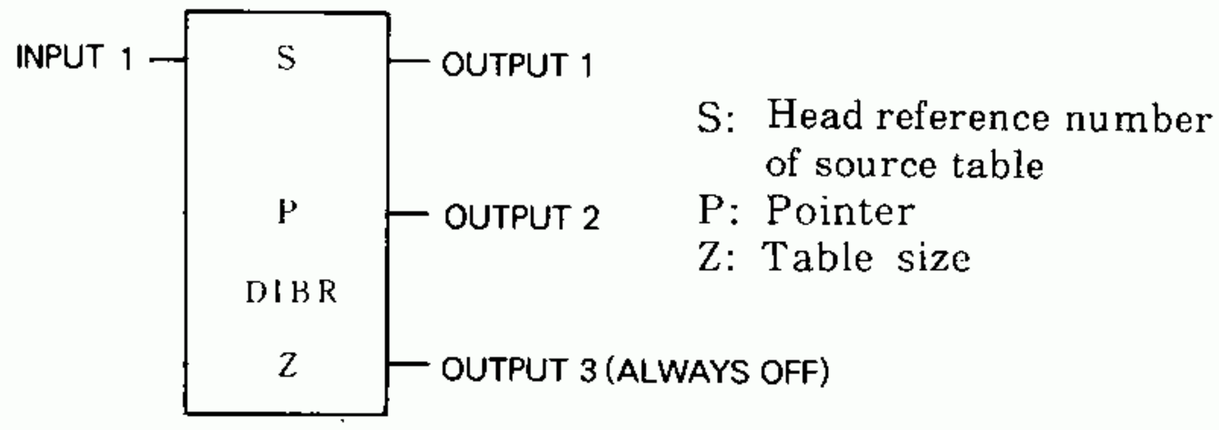


Fig. 5.68 DIBR General Form

- Fig. 5.68 shows the form of the block move 2 with destination index (DIBR).
- DIBR is the symbol denoting the block move 2 with destination index.
- DIBR requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.79, specify any of number for each of the elements.

Table 5.79 DIBR Elements

Element	Description	Specified Number
Top	Source table head reference number	Any of the following : • Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	With pointer, specified head number of destination table	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Size of source table and destination table	• For input relay specified in top, constant (1-32) • For link coil specified in top, constant (1-64) • Others (1-100)

1. When a coil and relay reference No. is to be specified in the top stage, the lower-place 4 digits of the number that can be specified will be limited to $[16\ m + 1]$. $(m = 0, 1, 2, \cdots)$
2. In case coil or relay is specified in the top stage when transfer of data is executed, data are stored in the registers in the sequence of the latest number starting with the highest order of register.

(3) Operation

- By DIBR, all data of the source table will be moved to the destination table (holding register groups) specified with the value i of the pointer when the input 1 is ON. The output 1 is turned on. The move will be completed in one scanning cycle.

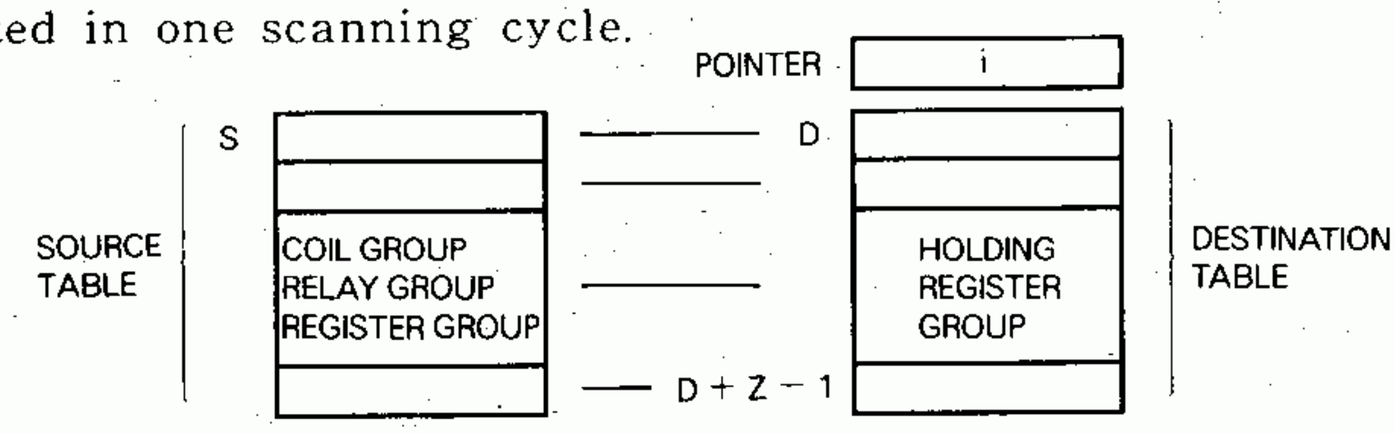
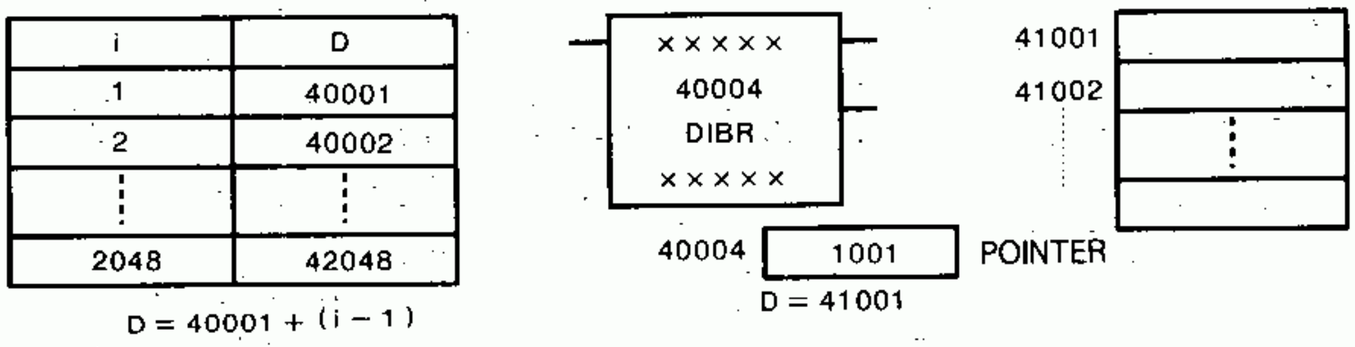


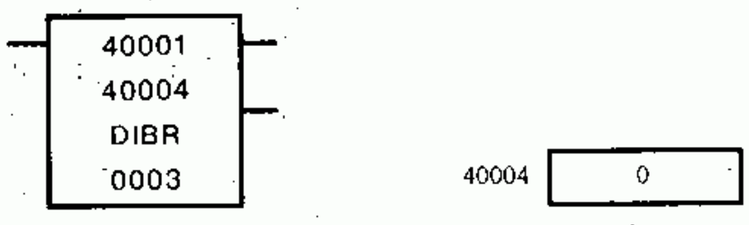
Fig. 5.69 DIBR Function

The following shows the relation between the value i of the pointer and the head number of the holding register group at the destination.



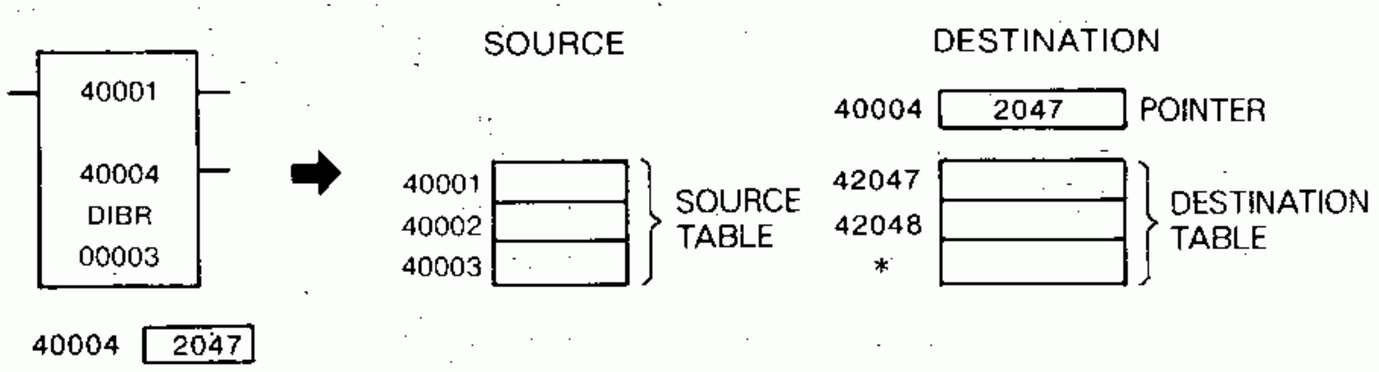
- In the following cases, the move will not be executed and the output 2 is turned on.

(a) When the value i of the pointer is out of the range of 1-2048



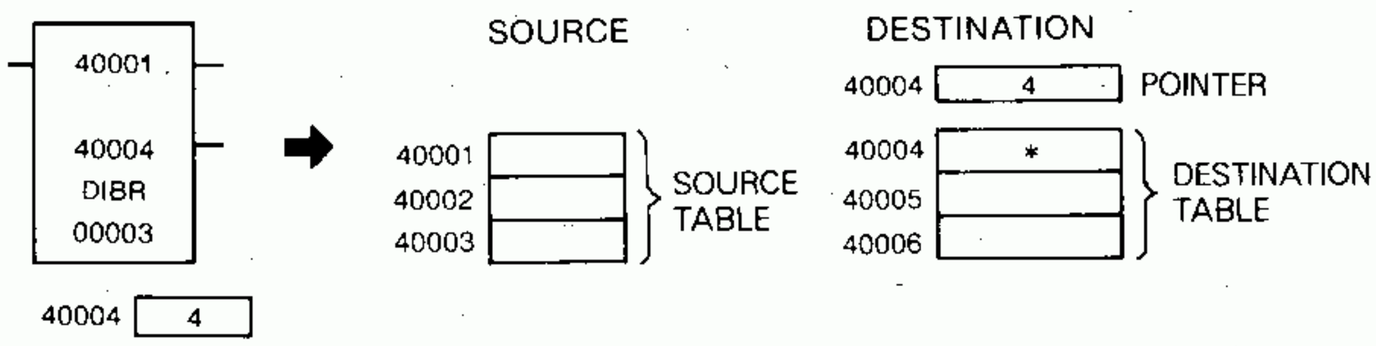
Note **There is no holding register group corresponding to i = 0. Therefore, no destination tables corresponding to the source tables 40001-40003 exist.**

(b) When the value i of the pointer is within the range of 1-2048, but no destination table to be specified by the value exist:



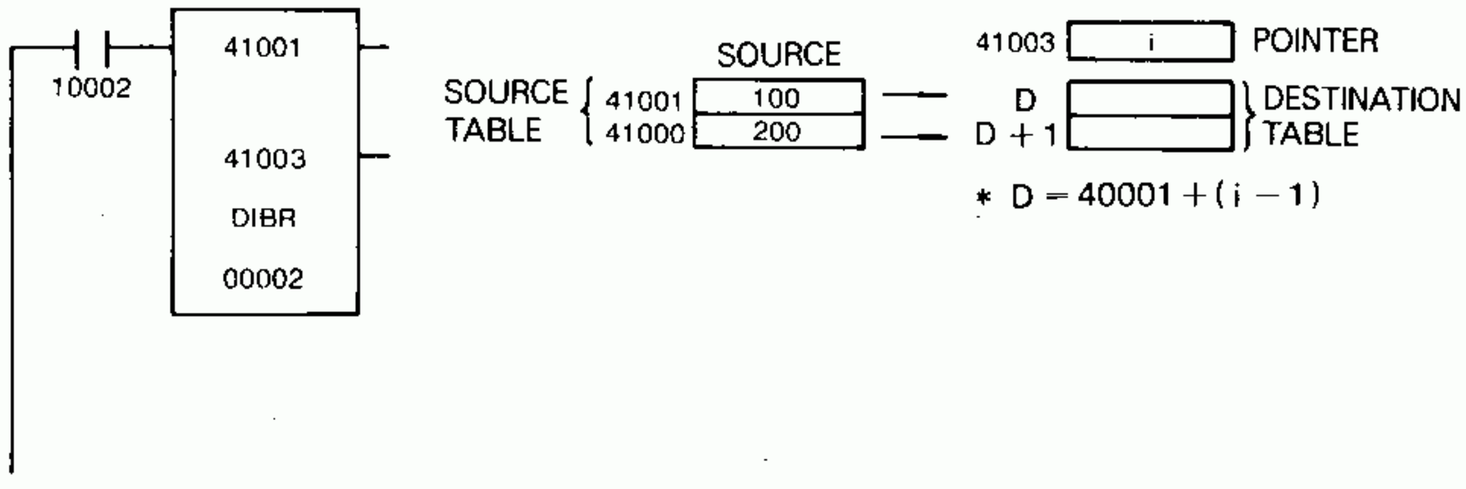
*No destination corresponding to the source 40003 exists.

(c) When the pointer is included in the destination table:

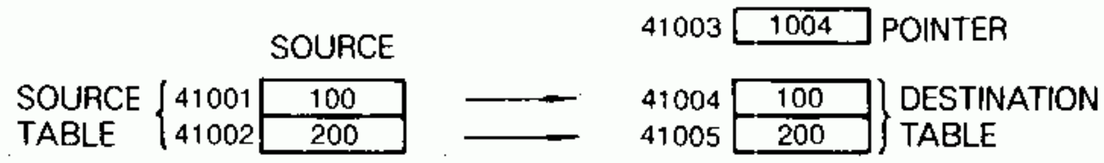


* The pointer 40004 is included in the destination table.

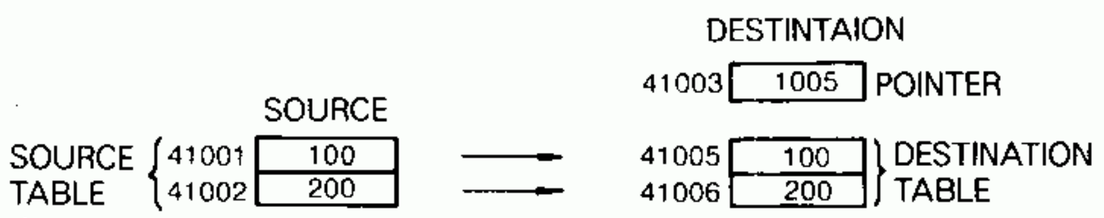
(4) Example



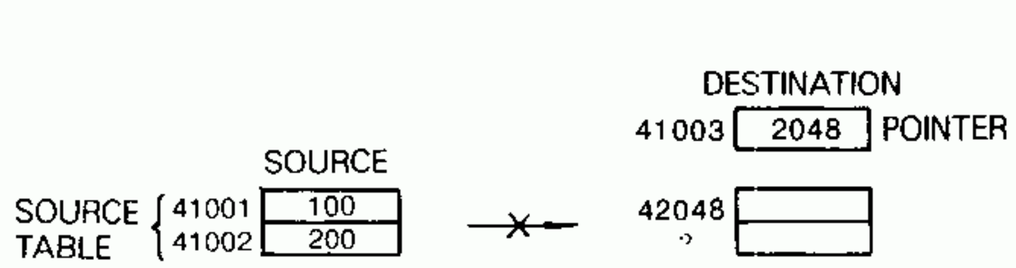
(a) When i = 1004, D1 = 41004. If the input relay 10002 is turned on at this time, the contents of the holding registers 41001 and 41002 is moved to the holding registers 41004 and 41005, and the output 1 is turned on.



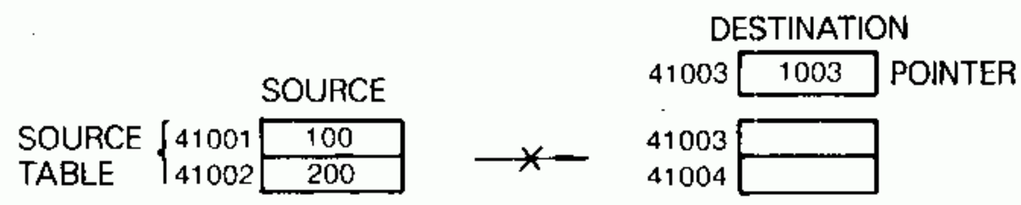
(b) When i = 1005, D1 = 41005. If input relay 10002 is turned on at this time, the contents of the holding registers 41001 and 41002 is moved to the holding registers 41005 and 41006, and the output 1 is turned on.



(c) When i = 2048, D1 = 42048.. Because the table size is 2, no holding register of destination corresponding to the source holding register 41002 exists. Therefore, even if the input relay 10002 is turned on, the move will not be performed and the output 2 is turned on.



(d) When i = 1003, D1 = 41003. Because the pointer 41003 is included in the destination table, the move will not be performed and the output 2 is turned on even if the input relay 1002 is turned on.



5.10.4 Block Move 1 with Source Index (SIBT)

(1) Function

This function transfers all the data of the source table (coil, relay, register group) specified by the pointer content (i) to the destination table (register group) in 1 scan.

(2) Form

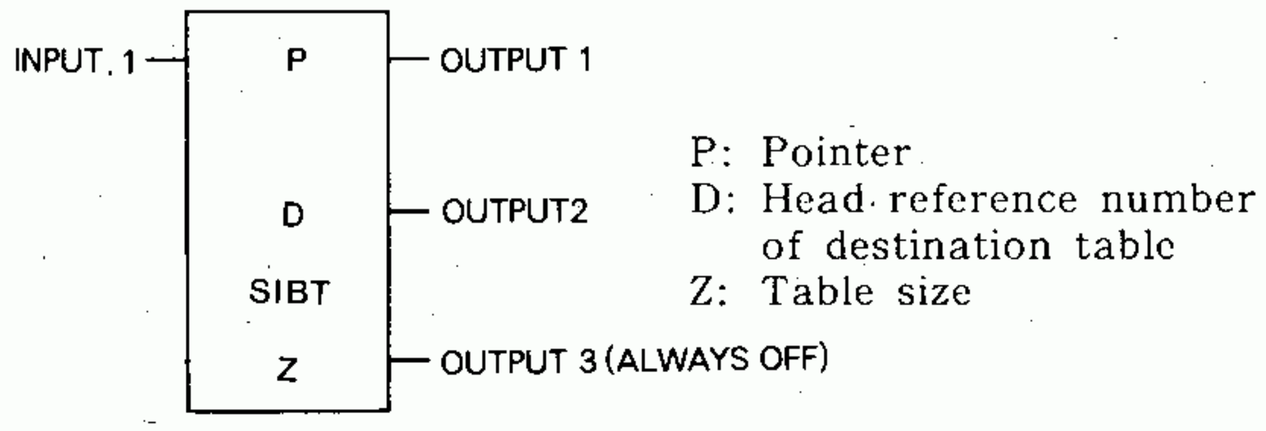


Fig. 5.70 SIBT General Form

- Fig. 5.70 shows the form of the block move 1 with source index (SIBT).
- SIBT is the symbol denoting the block move 1 with source index.
- **SIBT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.80, specify either constant K or reference number of various registers for each of the elements.**

Table 5.80 SIBT Elements

Element	Description	Specified Number
Top	With pointer, head number of the source table	• Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Head number of the destination table	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

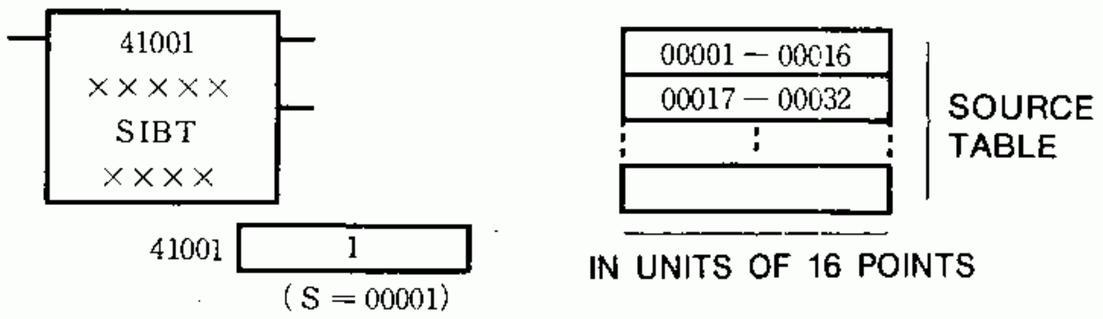
(3) Operation

By SIBT, all data of the source table (coil, input relay, input register group) specified by the value i of the pointer will be moved to the destination table (holding register groups) when the input 1 is ON. The output 1 is turned on. The move will be completed in one scanning cycle. Table 5.81 shows relation between the value i of the pointer and the source table (coil, input relay, input register group).

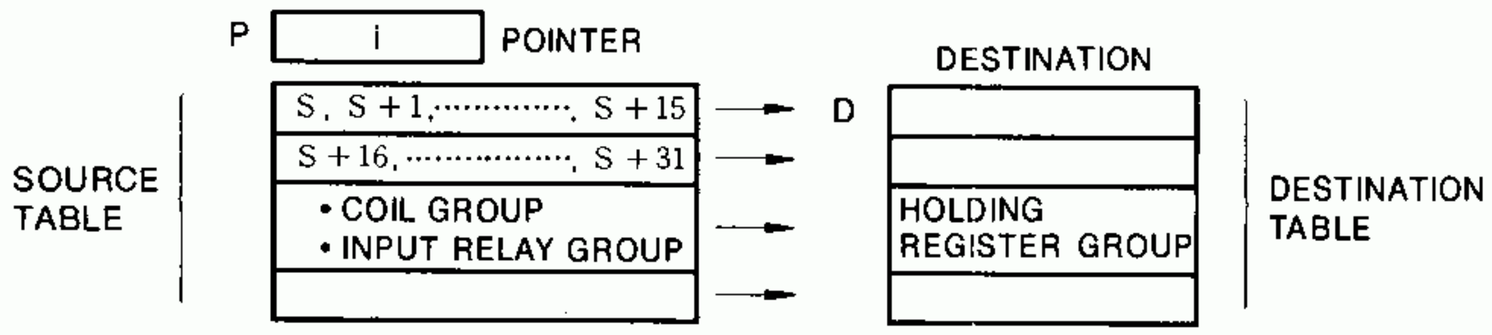
Table 5.81 Relation between Pointer i and Source Table

Coil		Input Relay		Input Register	
i	S	i	S	i	S
1	00001	1001	10001	3001	30001
2	00017	1002	10017	3002	30002
⋮	⋮	⋮	⋮	⋮	⋮
128	02033	1032	10497	3128	30128
S = 1+16(i-1)		S = 1001+16(i-1)		S = 30001+(i-3001)	

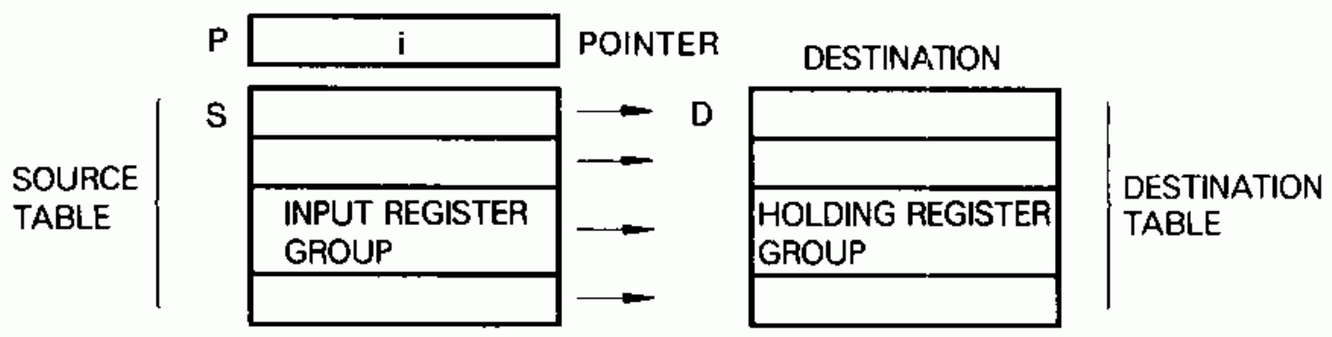
Example:



(a) When the source table is coil and input relay group:



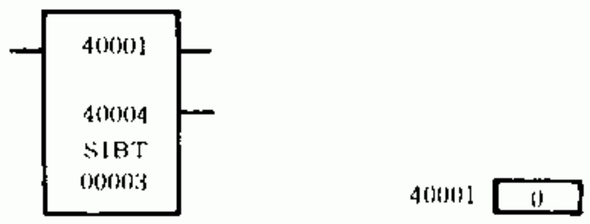
(b) When the source table is input register group:



- In the following cases, the move will not be executed and the output 2 is turned on.

(a) When the value i of the pointer is out of the ranges of 1-128, 1001-1032, and 3001-3218:

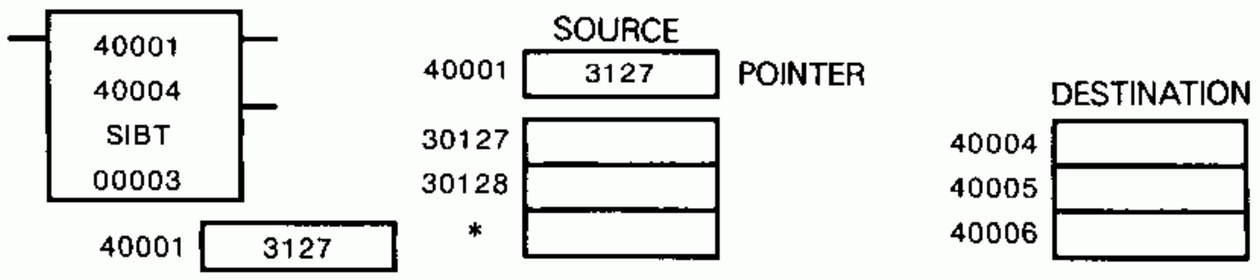
Example:



Note There are no coil group, input relay group and input registers corresponding to i = 0. Therefore, no source table corresponding to the destination tables 40004-40006 exist.

(b) When the value i of the pointer is within the ranges of 1-128, 1001-1032, or 3001-3128 but no source table specified by the value exists:

Example:

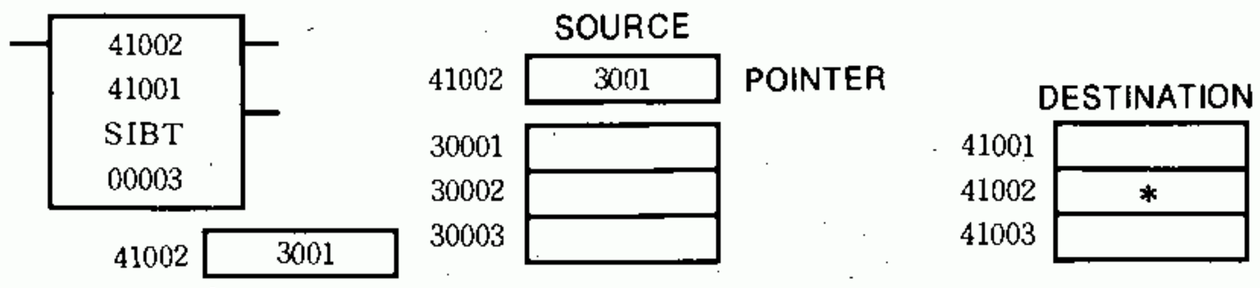


*No source input register corresponding to the destination 40006 exists.

5.10.4 Block Move 1 with Source Index (SIBT) (Cont'd)

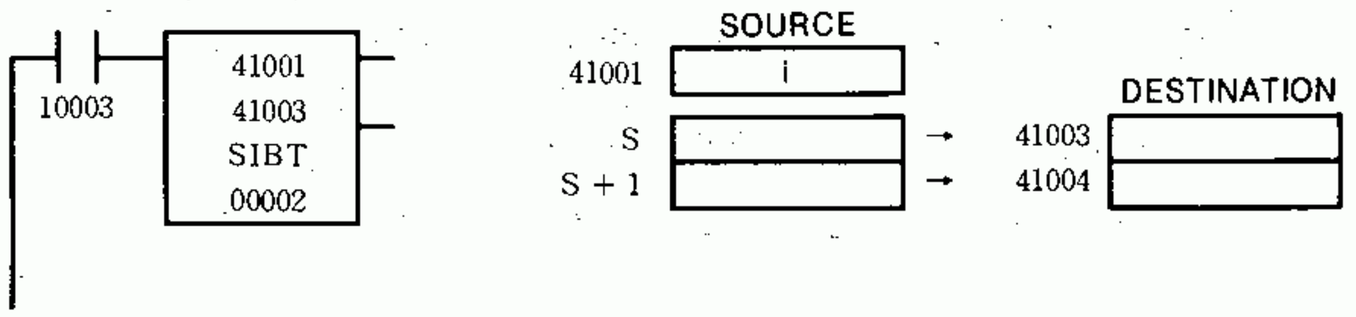
(c) When the pointer is included in the destination table:

Example:

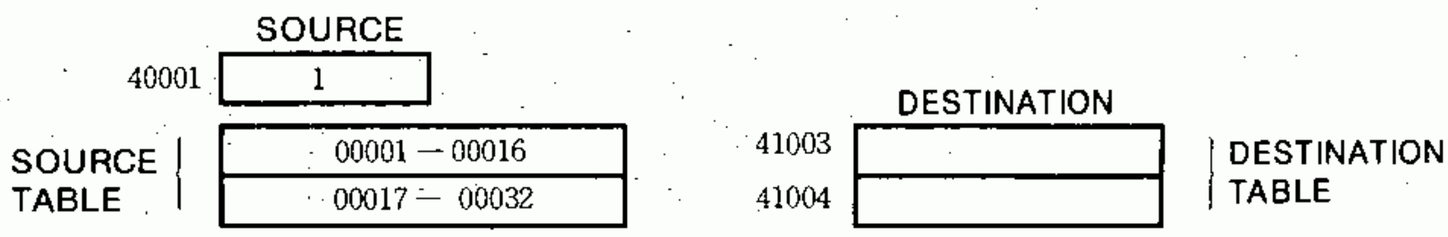


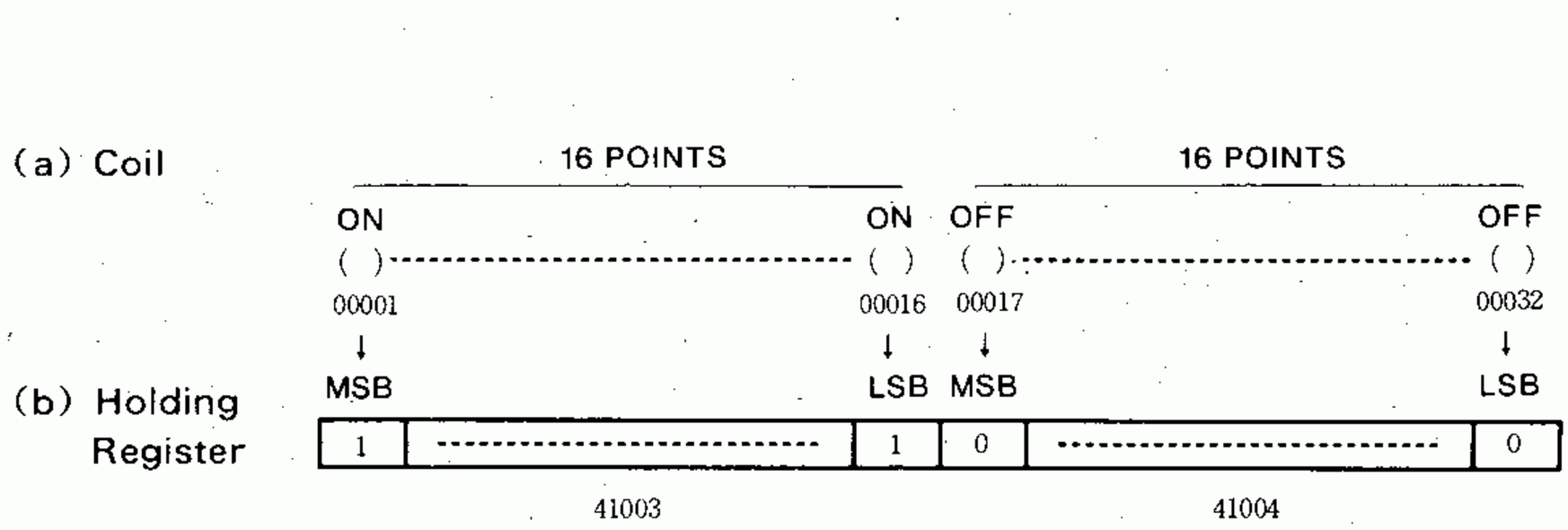
*The pointer 41002 is included in the destination table.

(4) Example

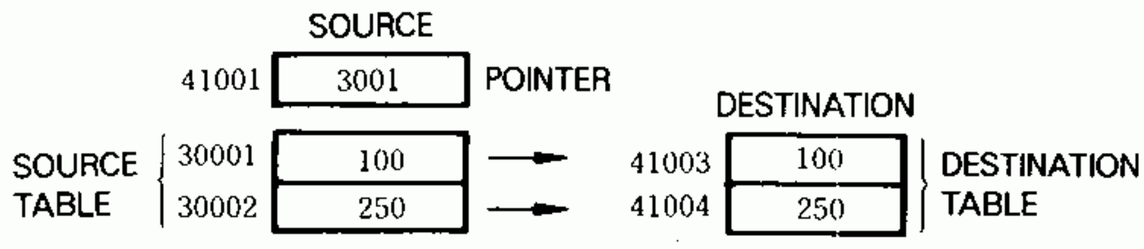


(a) When i = 1, S = 00001. If the input relay 10003 is turned on at this time, the ON/OFF status of the coil groups 00001-00032 is moved to the holding registers 41003 and 41004 and the output 1 is turned on.

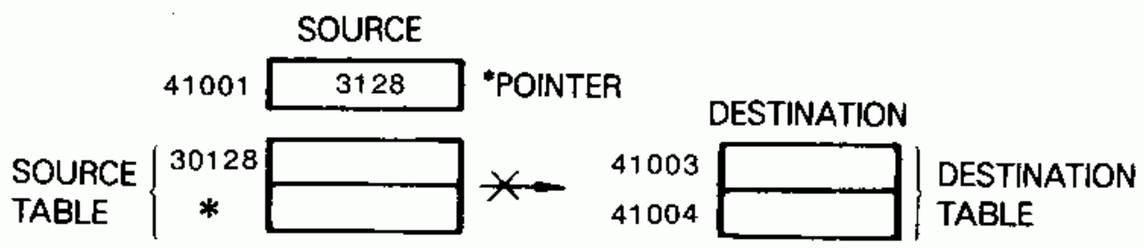




(b) When i = 3001, S = 30001. If the input relay 10003 is turned on at this time, the contents of the input registers 30001 and 30002 is moved to the holding registers 41003 and 41004, respectively.



(c) When i = 3128, S1 = 30128. Because the table size is 2, no source input register corresponding to the destination holding register 41004 exists. Therefore, even if the input relay 10003 is turned on, the move will not be performed and the output 2 is turned on.

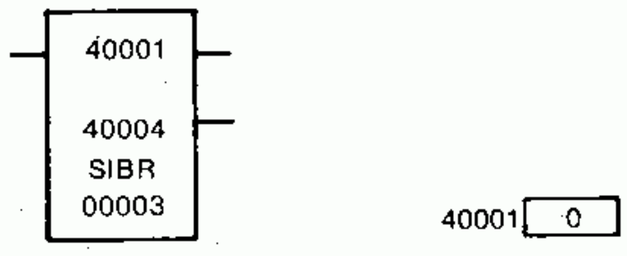


*No source input register corresponding to the destination 41004 exists.

- In the following cases, the move will not be executed and the output 2 is turned on.

(a) When the value i of the pointer is out of the range of 1-2048:

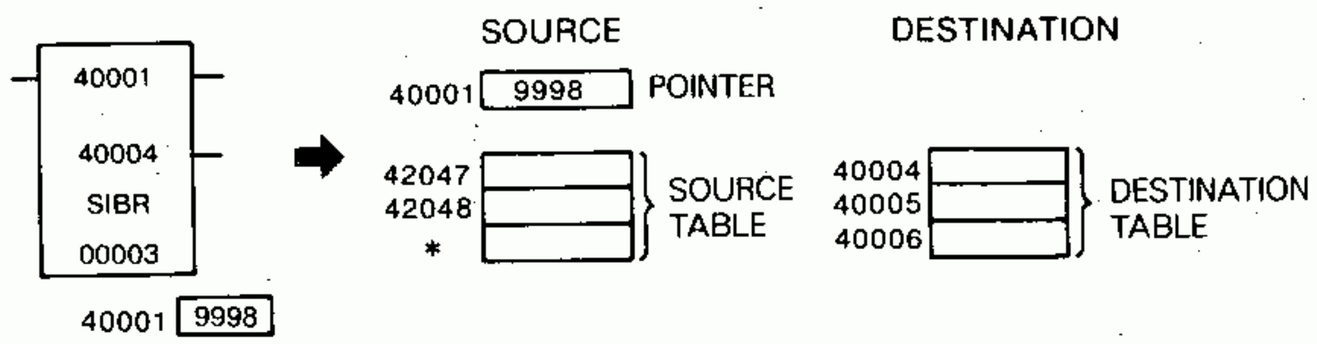
Example:



Note No holding register corresponding to i = 0 exists and therefore no source table corresponding to the destination table 40004-40006 exists.

(b) When the value i of the pointer is within the range of 1-2048 but no source table specified by the value exists:

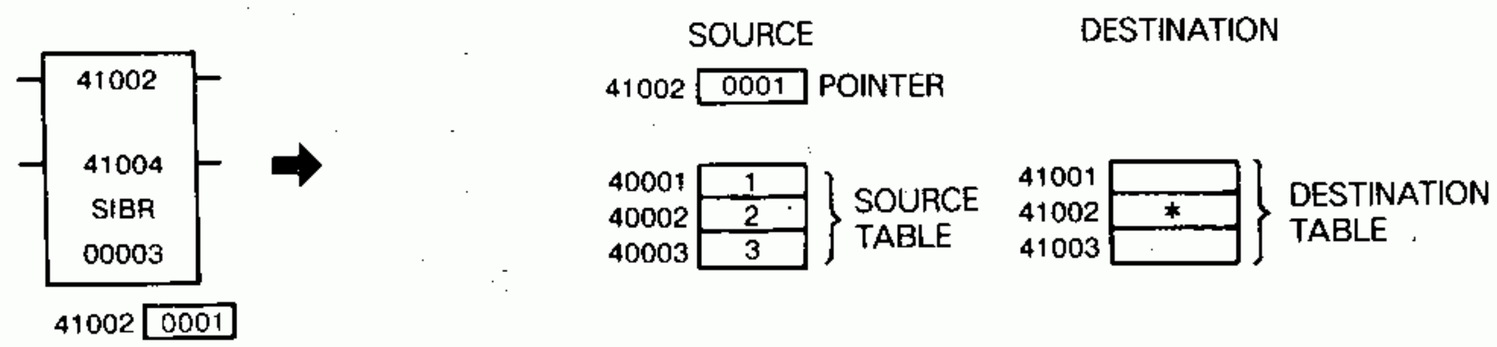
Example:



* No source holding register corresponding to the destination 40006 exists.

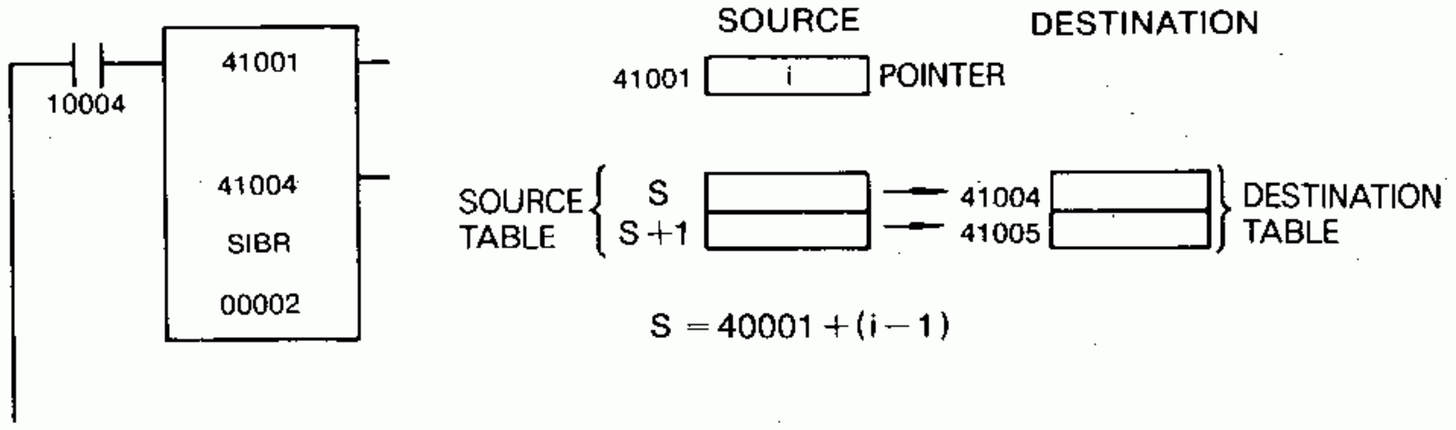
(c) When the pointer is included in the destination table:

Example:

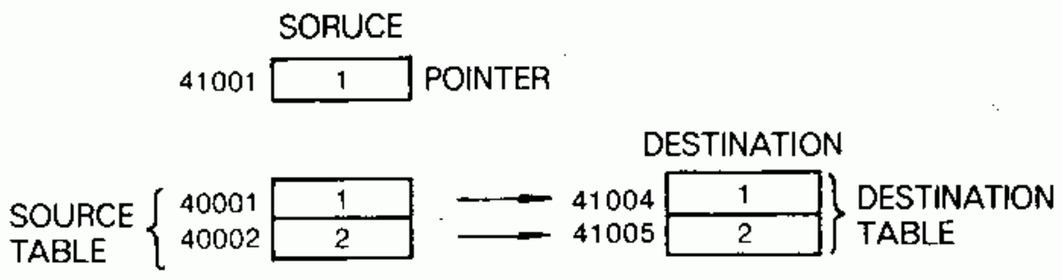


* The pointer 41002 is included in the destination table.

(4) Example



- When i = 1, S = 40001. If the input relay 10004 is turned on at this time, the contents of the holding registers 40001 and 40002 is moved to the holding registers 41004 and 41005 respectively, and the output 1 is turned on.



5.10.5 Block Move 2 with Source Index (SIBR)

(1) Function

This function can transfer all data of source table (holding register group) specified by the value i of the pointer to the destination table (holding register group).

(2) Form

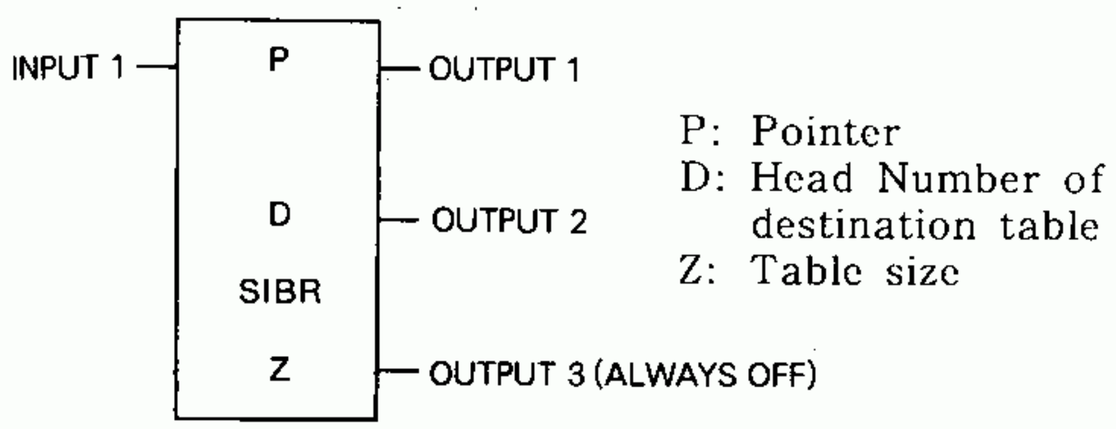


Fig. 5.71 SIBR General Form

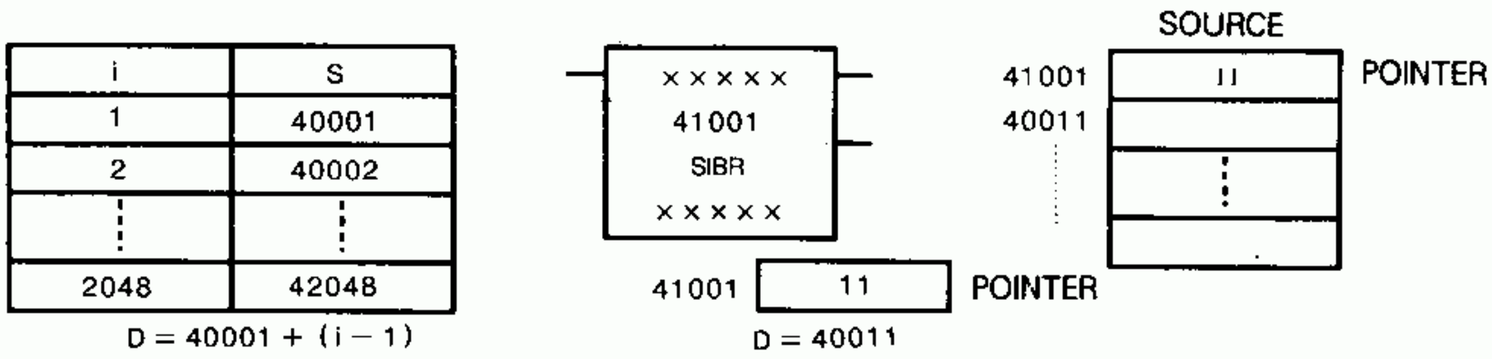
- Fig. 5.71 shows the form of block move 2 with source index (SIBR).
- **SIBR is the symbol denoting the block move 2 with source index.**
- SIBR requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.82, specify the needed number for each of the elements.

Table 5.82 SIBR Elements

Element	Description	Specified Number
Top	With pointer, specified head number of source table.	• Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Head number of destination table	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	Constant (1-100)

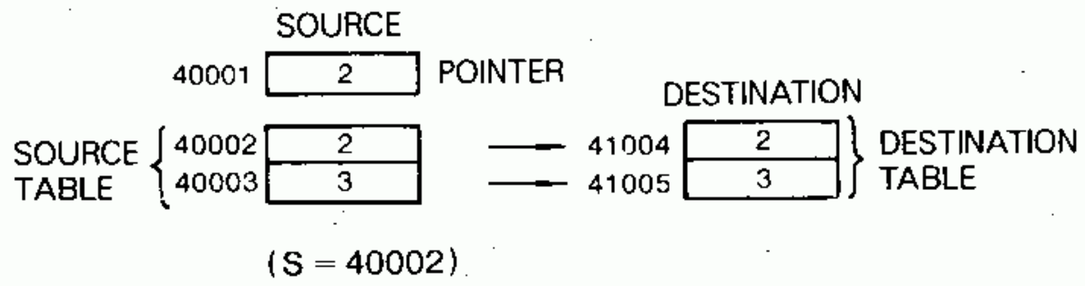
(3) Operation

The following figure shows relation between the value i of the pointer and the head number of source table (holding register group).

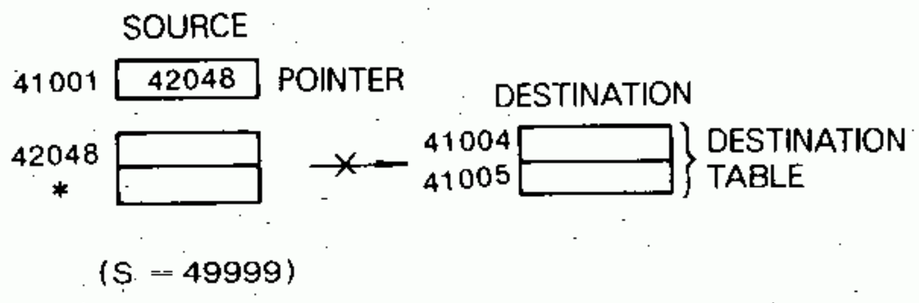


5.10.5 Block Move 2 with Source Index (SIBR) (Cont'd)

- When i = 2, S = 40002. If the input relay 10004 is turned on at this time, the block move will be performed as follows.



- When i = 2048, S = 42048. Because the table size is 2, no holding register corresponding to the destination 41005 exists. Therefore, even if the input relay 10004 is turned on, the move will not be performed and the output 2 is turned on.



*No source holding register corresponding to the destination 41005 exists.

5.10.6 Programming Indexed Block Move Circuit and Precautions

(1) Inputs to the indexed block move circuit may be outputs of relays, timers, counters, arithmetic operations, data transfer matrixes, and other indexed block move circuits.

(2) Coils need not be connected to two output nodes, (1 and 2) of an indexed block move circuit. It is permitted to connect a relay contact to the output nodes at right or connect the ouput node directly to an input node of an arithmetic circuit, except relays.

(3) To execute the move constantly, connect the input directly to the power rail at left. To execute it only during one scanning cycle, use a transitional contact as an input.

(4) The range of the source or destination table specified by a table size must be within the range of the reference numbers of input relays, coils, or registers.

(5) It is possible to OR the outputs by connecting a vertical shunt element.

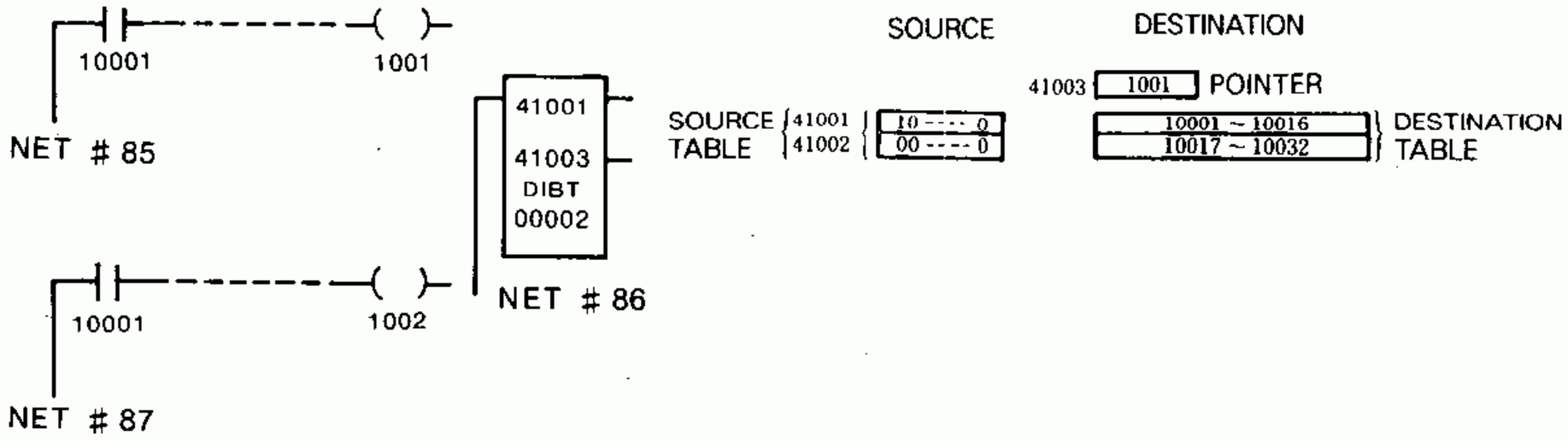
(6) The source or the pointer remain unchanged data or values after the move.

(7) Be sure to disable the input realy groups used as destination of the block move 1 with destination index (DIBT) (DISABLE ON or DISABLE OFF). There are following differences between the cases the input relay groups are disabled and enabled.

(a) When the input relay groups are disabled:
The input relays are turned on and off according to the results of execution of DIBT.

(b) When the input relay groups are enabled:
The input relays are turned on and off accroding to the results of execution of DIBT then all of them are turned off at the beginning of the next scanning cycle. But, if an input module is connected and I/O allocation is made, they are turned on and off according to the actual status of input signals at the beginning of the next scanning cycle.

Example:



Note: If the input relay 10001 is disabled, the each normally open (NO) contact of the input relay 10001 of NET #85 and #87 is turned on. If the input relay 10001 is enabled, the NO contact of the input relay 10001 of NET #85 is turned off and that of NET #87 is turned on.

(8) Do not use transitional contacts of input relay group used in DIBT destination. Because of the followings:

(a) $\dashv\uparrow\vdash_{1\times\times\times}$ provides a function of $\dashv\vdash_{1\times\times\times}$.

(b) $\dashv\downarrow\vdash_{1\times\times\times}$ keeps always OFF.

5.11 DATA CONVERSION

Data conversion is designed to swap, sort, compose, split and/or convert the data in the data table.

5.11.1 Types of Data Conversion

There are 7 types of data conversion as follows:

Table 5.83 Types of Data Conversion

Type	Symbol	Functions	Reference Page
BCD→Binary Conversion	BIN	Converted to binary	181
Binary→BCD Conversion	BCD	Converted to BCD	184
Swap	SWAP	Replacement of high-order byte and low-order byte.	187
Sort	SORT	Rearrangement of data in UP order and DOWN order.	189
Byte Split	BYSL	Splitting of word data into byte data.	192
Byte Composition	BYCM	Composition of byte data into word data.	194
Block Addition	BADD	Addition in block unit.	196

5.11.2 BCD → Binary Conversion (BIN)

(1) Function

This function converts the content of source table represented in BCD into a binary number in 1 scan cycle and transfers it to the destination.

(2) Form

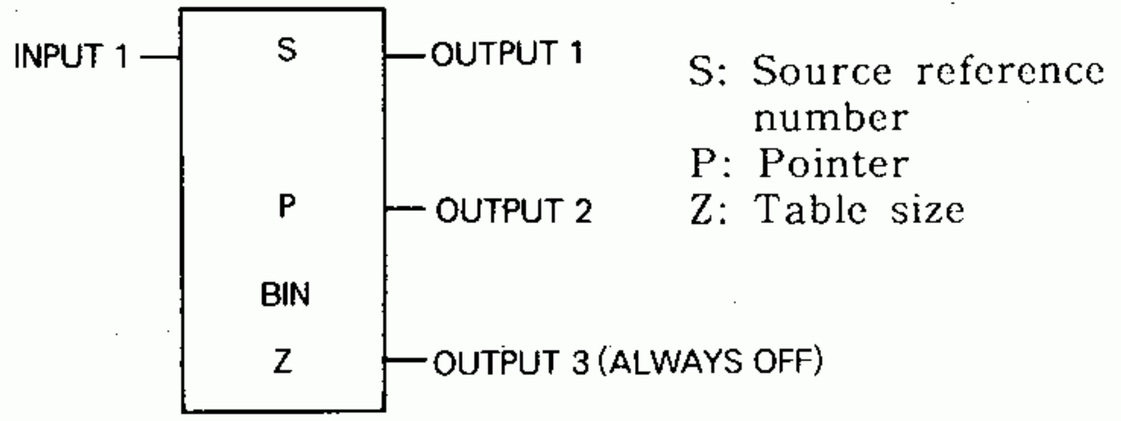


Fig. 5.72 BIN General Form

- Fig. 5.72 shows the form of BCD → BIN conversion.
- BIN is the symbol denoting BCD → BIN conversion.
- BCD → BIN conversion requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.84, specify the needed number for each element.

Table 5.84 BIN Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Pointer	• Holding register (40001-42048) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-16)

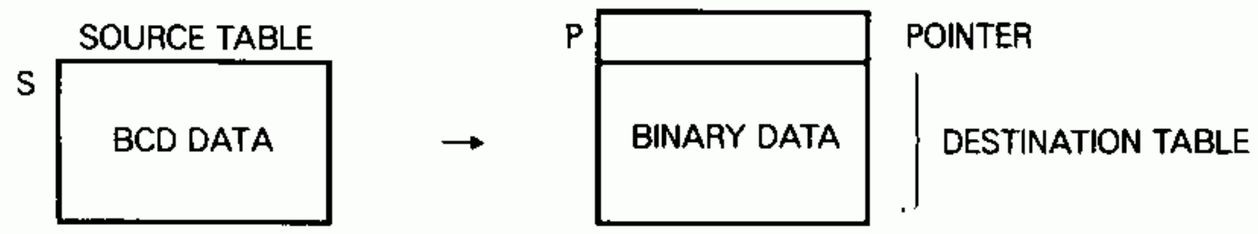
Note 1. Destination reference number starts with the next to pointer reference number.
2. Table size does not contain the pointer.

(3) Operation

When the input 1 is in the ON state, this function converts all of the data of source table from BCD into binary number and transfers it to the destination table.

When the data of source table are not BCD data, the function indicates the order of that data counted from the head as the order counted from below the pointer (or from LSB). The conversion is perfomed down to the end.

The data which are not BCD data (the number exceeding 9999) are tentatively converted as BCD data into a binary number and stored in the destination table, but the data are not correct.



If the data 3rd from the head of source table are not BCD data, the content of the pointer after the conversion will be as follows:

	MSB															LSB
POINTER	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

The output 1 will become ON when the input 1 is ON and will become OFF (Copy of the input 1) when the input 1 is OFF.

The output 2 will become ON when there are any data other than BCD data in the source table.

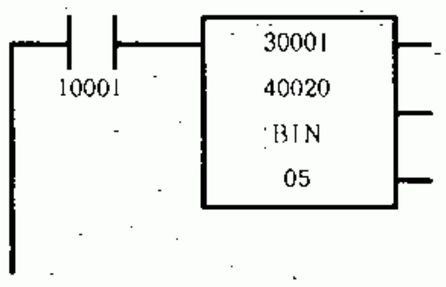
The output 3 is always OFF.

The operations of BIN are tabulated in Table 5.85.

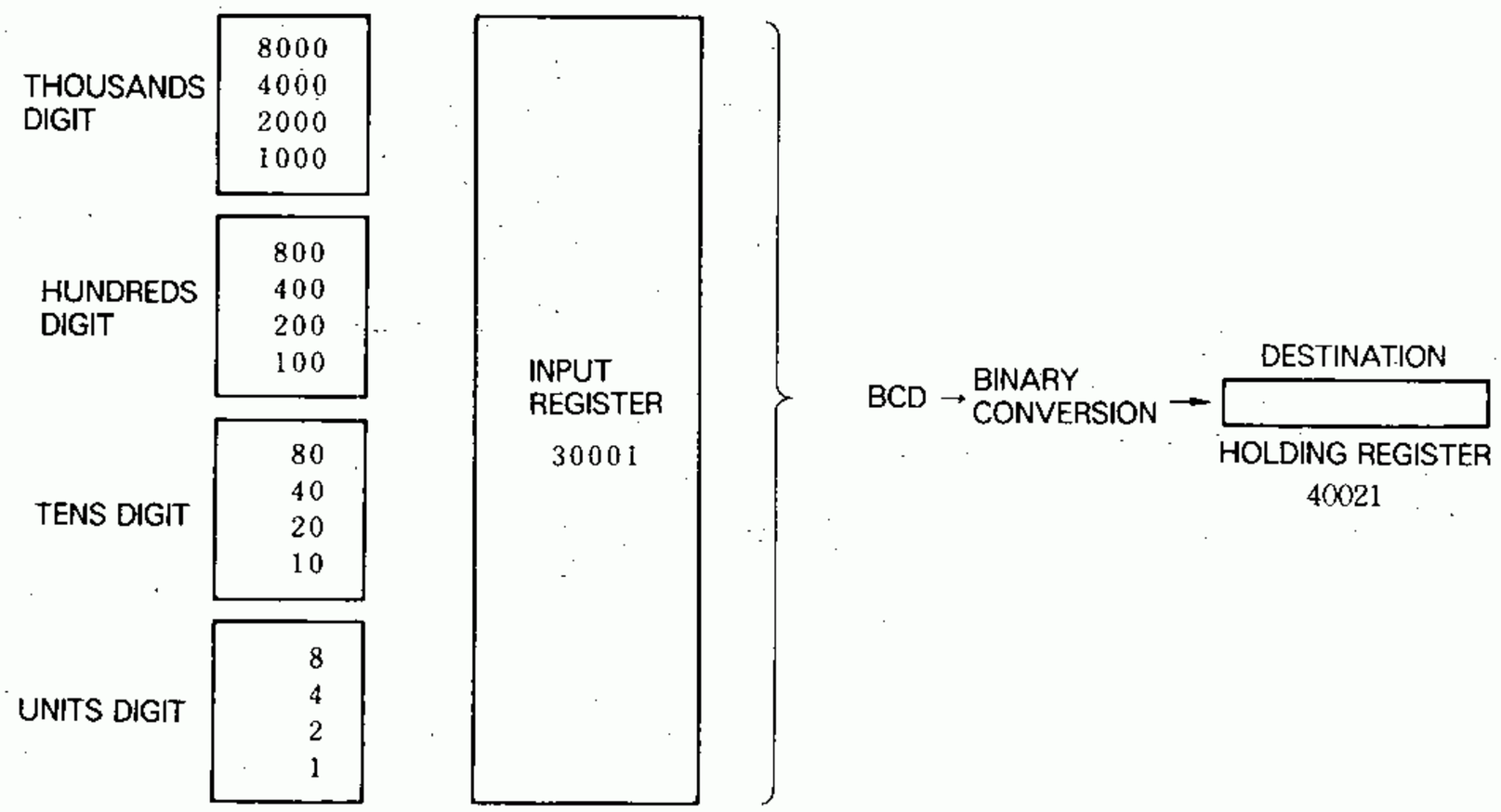
Table 5.85 BIN Operations

Input 1	Operations		Output 1	Output 2
ON	No.15 bit (8000) No.14 bit (4000) No.13 bit (2000) No.12 bit (1000) No.11 bit (800) No.10 bit (400) No.9 bit (200) No.8 bit (100) No.7 bit (80) No.6 bit (40) No.5 bit (20) No.4 bit (10) No.3 bit (8) No.2 bit (4) No.1 bit (2) No.0 bit (1)	BCD→binary conversion	ON	OFF
	Converted (other type of data in source table...not BCD data)		ON	ON
OFF	Not operated.		OFF	OFF

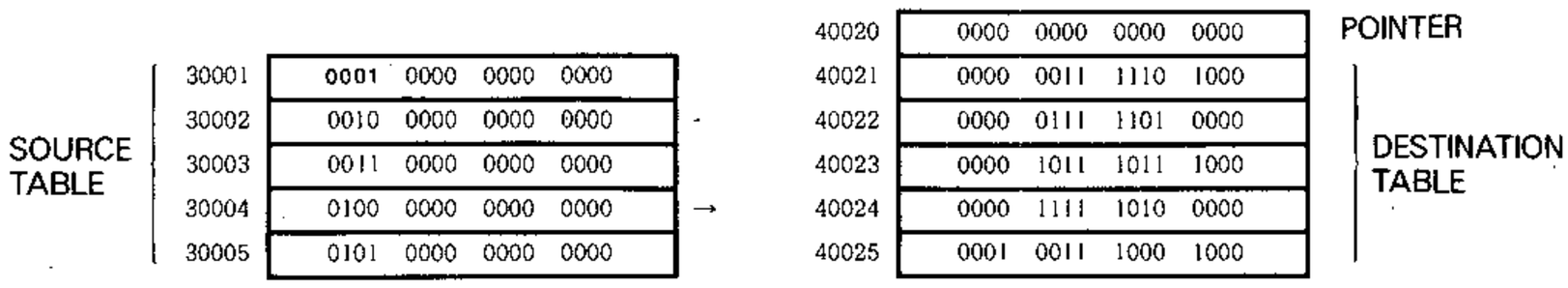
(4) Example



(a) Ladder



(b) Input Conversion (using 30001)



(c) Content of Conversion

BIN shown in (a) will execute the conversion shown in (C) throughout the duration when input relay 10001 is in the ON state.

BIN is mainly used to convert BCD data in the input register that are input, if the input device is a device of BCD representation.

Suppose that the data of 30004 are not BCD data, 0110 1111 0101 0000, this function will place 1 in the 4th bit counted from below 40020 of the pointer and continue conversion down to the last data of the source table. The data of 40024 will be tentatively converted into a binary number, but the data will not be correct. From the value of the pointer, it can be determined in which register of the source table the data which are not BCD data are stored. As described above, when certain data which are not BCD data are included in the source table, the output 2 will turn ON.

5.11.3 Binary → BCD Conversion (BCD)

(1) Function

This function converts the content of source table represented in binary into a BCD in 1 scan and transfers it to the destination.

(2) Form

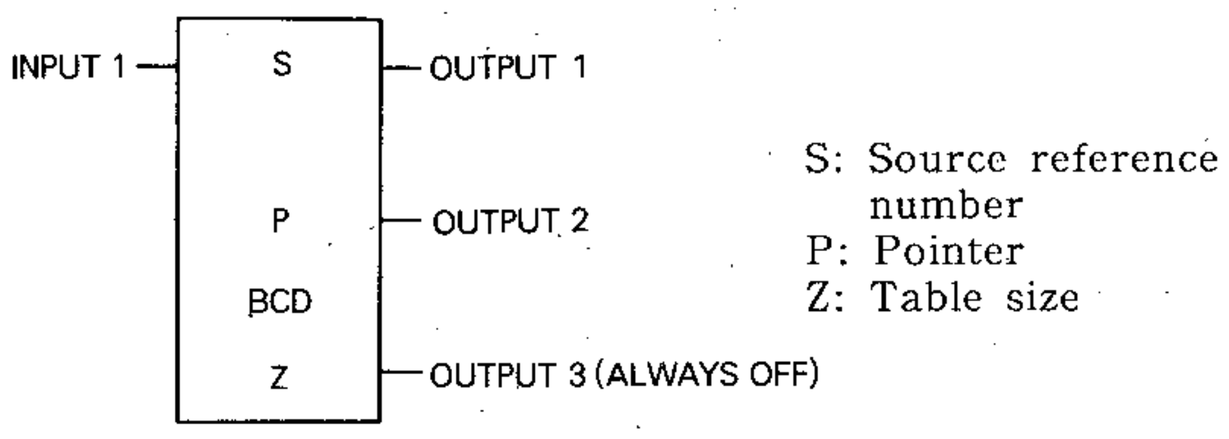


Fig. 5.73 BCD General Form

- Fig. 5.73 shows the form of Binary → BCD conversion.
- BCD is the symbol denoting Binary → BCD conversion.
- Binary → BCD conversion requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.86, specify the needed number for each element.

Table 5.86 BCD Elements

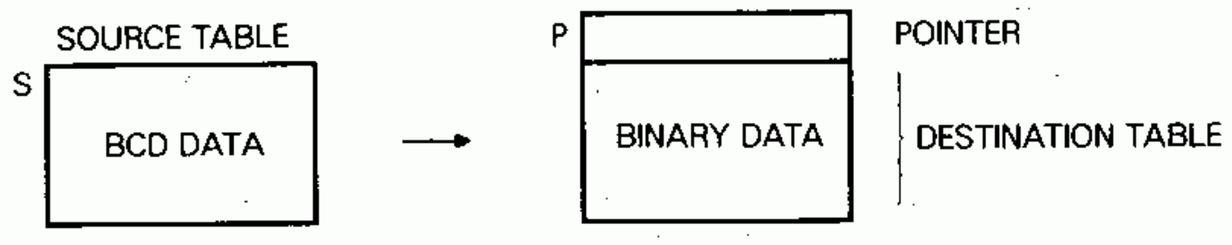
Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Pointer	• Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-16)

Note 1. Destination reference number starts with the next to pointer reference number.
2. Table size does not contain the pointer.

(3) Operation

When the input 1 is in the ON state, this function converts all of the data of source table from Binary into BCD and transfers it to the destination table.

When the data of source table are not of BCD conversion range (the number exceeding 9999 or negative number), the function indicates the order of that data counted from the head as the order counted form below the pointer (or from LSB). The conversion is peformed down to the end. The data are tentatively converted into a BCD and stored in the destination table, but the data are not in BCD.



If the data 3rd from the head of source table are not binary table, the content of the pointer after the covnersion will be as follows:

	MSB															LSB
POINTER	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

The output 1 will become ON when the input 1 is ON and will become OFF (Copy of the input 1) when the input 1 is OFF.

The output 2 will become ON when there are any data other than binary data in the source table.

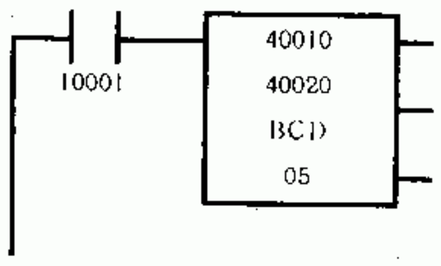
The output 3 is always OFF.

The operations of BCD are tabulated in Table 5.87.

Table 5.87 BCD Operations

Input 1	Operations		Output 1	Output 2
ON	Binary → BCD Conversion	No.15 bit (8000) No.14 bit (4000) No.13 bit (2000) No.12 bit (1000) No.11 bit (800) No.10 bit (400) No.9 bit (200) No.8 bit (100) No.7 bit (80) No.6 bit (40) No.5 bit (20) No.4 bit (10) No.3 bit (8) No.2 bit (4) No.1 bit (2) No.0 bit (1)	ON	OFF
	Converted (other type of data in source table...not binary data)		ON	ON
OFF	Not operated.		OFF	OFF

(4) Example



(a) Ladder

SOURCE TABLE

40010	1000	0000	1010	0000	(−160)
40011	0000	1000	0000	0000	(2048)
40012	0000	0000	0000	1111	(15)
40013	0000	0000	0001	0000	(16)
40014	0000	0001	0000	0000	(256)

DESTINATION

40020	0000 1000 0000 0000	POINTER
40021	ANY PATTERN	
40022		
40023		
40024		
40025		

① Before Conversion

SOURCE TABLE

40010	1000	0000	1010	0000	(−160)
40011	0000	1000	0000	0000	(2048)
40012	0000	0000	0000	1111	(15)
40013	0000	0000	0001	0000	(16)
40014	0000	0001	0000	0000	(256)

DESTINATION

40020	0000 0000 0000 0001	POINTER
40021	WRONG DATA	
40022	0010 0000 0100 1000	
40023	0000 0000 0001 0101	
40024	0000 0000 0001 0110	
40025	0000 0010 0101 0110	

② After Conversion

(b) Content of Conversion

BCD shwon in (a) will execute the conversion shown in (b) throughout the duration when input relay 10001 is in the ON state.

BCD is mainly used to output the internal operation result after converting its binary data into BCD, if the output device is a device of BCD representation.

Since the content of 40010 in the source table (A negative number is anticipated for this data) is, if converted into BCD, a 5-digit data of 32928 which does not come into the register, incorrect data will be stored. In this case, the function places 1 in the least significant bit of pointer 40020 and continues the conversion down to the last data of the source table. From the value of the pointer, it can be determined in which register of the source table the wrong data are stored. As described above, when any data which cannot be converted into a 4-digit BCD are included in the source table, the output 2 will turn ON.

5.11.4 Swap (SWAP)

(1) Function

It stores the content of source rable in the destination, after replacing the high-order byte with the low-order byte.

(2) Form

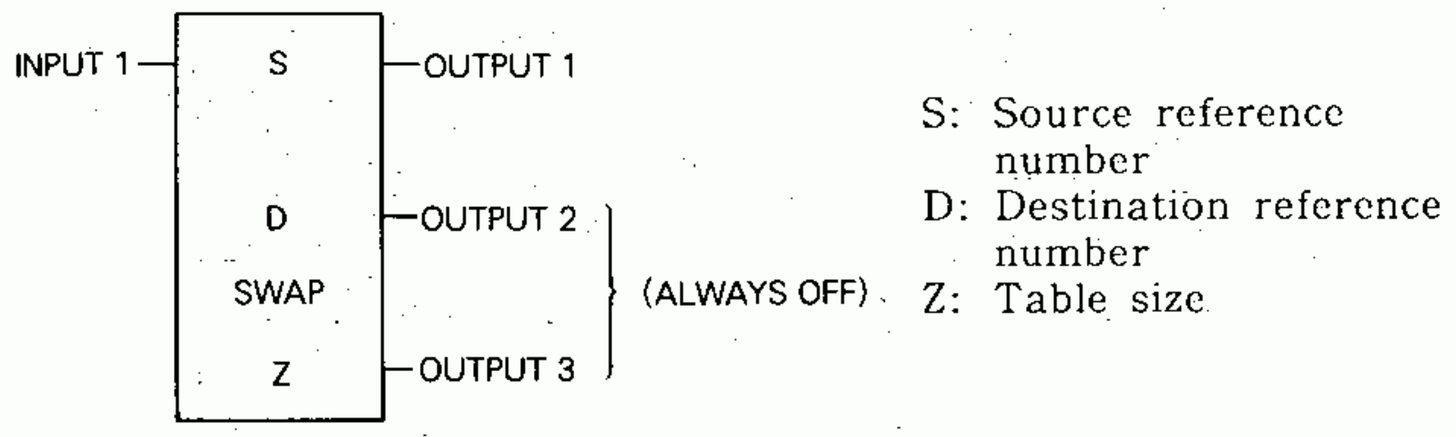


Fig. 5.74 SWAP General Form

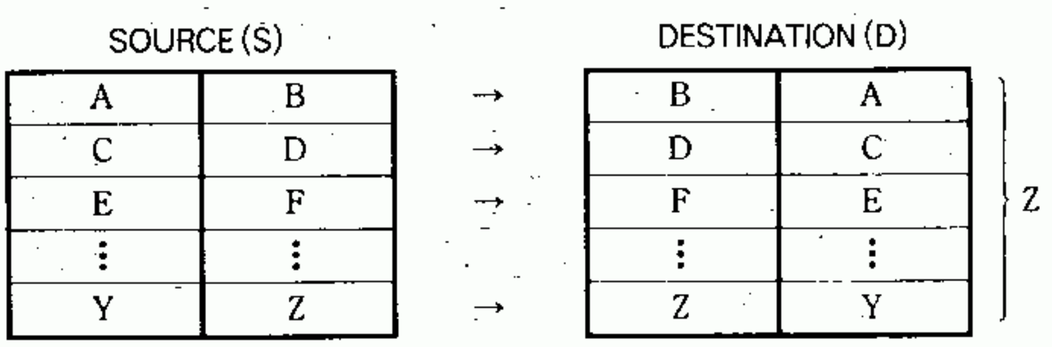
- Fig. 5.74 shows the form of swap.
- SWAP is the symbol denoting swap.
- SWAP requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.88, specify the needed number for each element.

Table 5.88 SWAP Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

(3) Operation

When the input is ON, the function stores the content of source table after replacing the high-order byte with the low-order byte. It converts and transfers the whole table in 1 scan cycle.



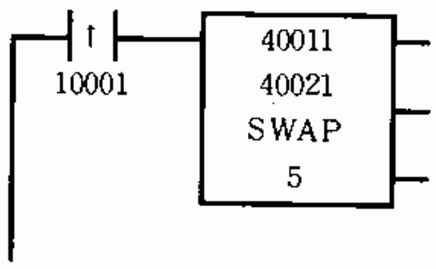
(a) Inputs

The input 1 executes SWAP and stores the result in the destination table. The inputs 2 and 3 are not used.

(b) Outputs

The operation of the output 1 is the same as the input 1 ON/OFF status. The outputs 2 and 3 are always OFF.

(4) Example



(a) Ladder

SOURCE

40011	0101	1111	0000	1111
40012	1111	1111	0000	0000
40013	1010	1010	0101	0101
40014	0001	1000	0100	0010
40015	1000	1100	0011	0011

DESTINATION

40021 40032 40023 40024 40025	ANY PATTERN

① Before Swap

SOURCE

40011	0101	1111	0000	1111
40012	1111	1111	0000	0000
40013	1010	1010	0101	0101
40014	0001	1000	0100	0010
40015	1000	1100	0011	0011

DESTINATION

40021	0000	1111	0101	1111
40022	0000	0000	1111	1111
40023	0101	0101	1010	1010
40024	0100	0010	0001	1000
40025	0011	0011	1000	1100

② After Swap

(b) Content of Swap

This function is effective for data rearrangement in use of Memobus data communication.

5.11.5 Sort (SORT)

(1) Function

It stores the content of source table in the destination table after rearranging it either in the order from the largest value down or in the order from the smallest value up. Or, with the source table as an index, it rearranges the destination table in the order from the largest value down or on the order from the smallest value up.

(2) Form

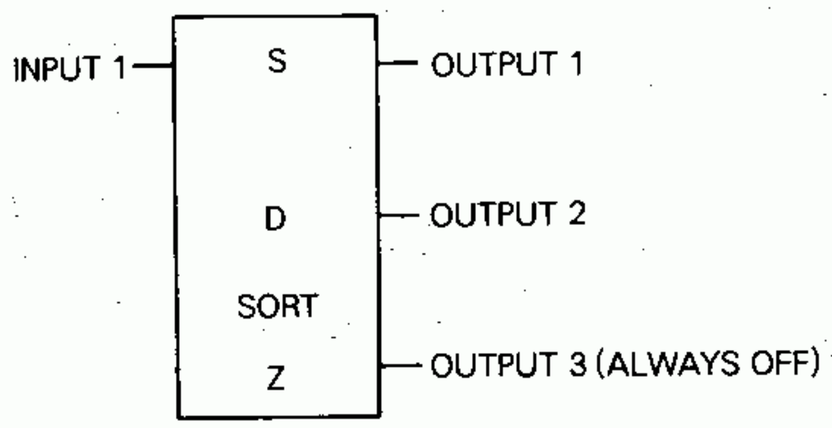


Fig. 5.75 SORT General Form

- Fig. 5.75 shows the form of sort.
- SORT is the symbol denoting sort.
- SORT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.89, specify the needed number for each element.

Table 5.89 SORT Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

Note
1. If the input 3 is OFF; when source table is equal to destination table, the operation can be performed correctly. However, when they overlap, no correct operation can be performed.
2. If the input 3 is ON; when source table and destination table overlap, no correct operation can be performed. When source table is equal to destination table, the output 2 turns on, and no operation is performed.

(3) Operation

The function transfers the content of source table to the destination table after sorting it either in the order from the largest value down or in the order from the smallest value up. Or, it sorts the destination table in the order from the largest value down or in the order from the smallest value up, and at the same time, it sorts the source table in the same way.

It rearranges all tables in 1 scan. Note that the table cannot be sorted correctly when there is any negative number in the data.

(a) Inputs

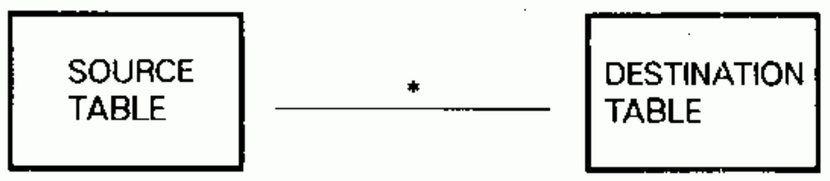
- Input 1: Executes SORT when it is ON.
- Input 2: Sorts the data in the order from the largest value down when it is ON and in the order from the smallest value up when it is OFF.
- Input 3: When it is ON, it makes destination table the subject of SORT using source table as an index. When it is OFF, it makes the content of source table the subject of SORT and transfers the result of SORT to the destination table.

(b) Outputs

- Output 1: Same as the status of the input 1. (Copy of the input 1)
- Output 2: It is ON when the input 3 is ON and source table is same as destination table, and in this case, no operation is performed.
- Output 3: Always OFF.

When the input 3 is ON:

When the input 3 is OFF:



*Data are stored in the order from the smallest value up or in the order from the largest value down.

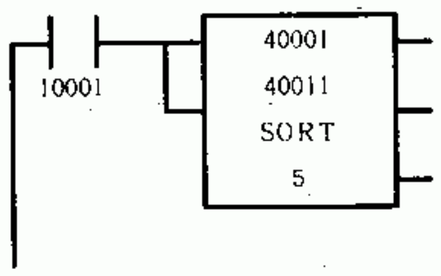
The operations of SORT are tabulated in Table 5.90.

Table 5.90 SORT Operations

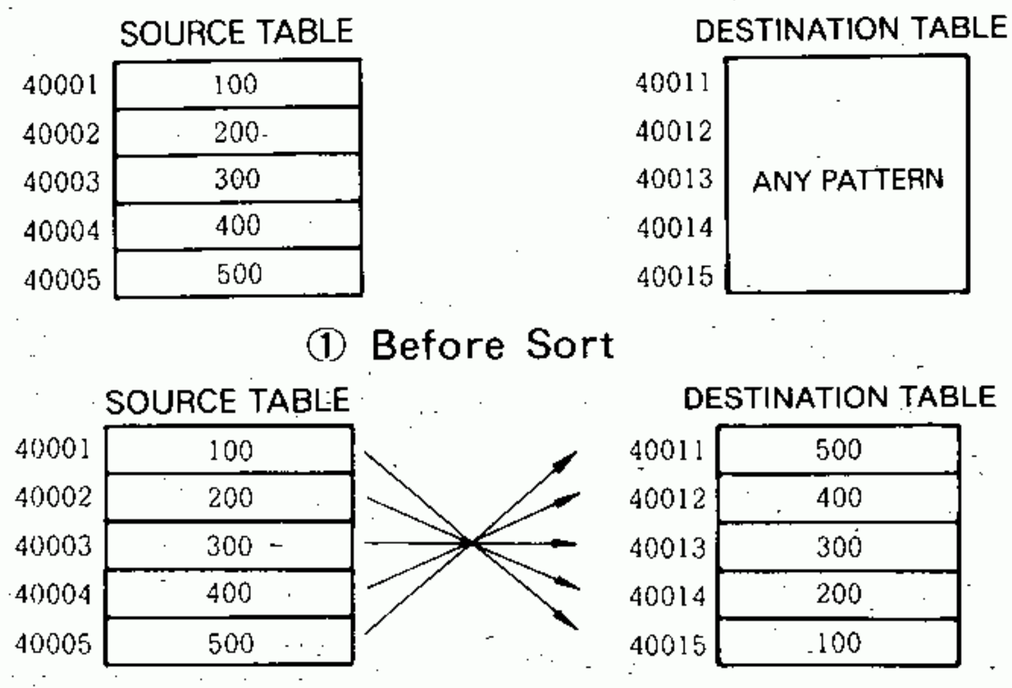
Input 1	Input 2	Input 3	Operations	Output 1	Output 2
ON	ON	ON	Indexed SORT in the order from the largest value down.	ON	OFF
		OFF	Transfers data to the destination after sorting in the order from the largest value down.		
	OFF	ON	Indexed SORT in the order from the smallest value up.		
		OFF	Transfers data to the destination in the order from the smallest value up.		
	—	ON	SORT and destination table coincide with each other.	ON	ON
OFF	—	—	Not operated.	OFF	OFF

(4) Example

Example 1:



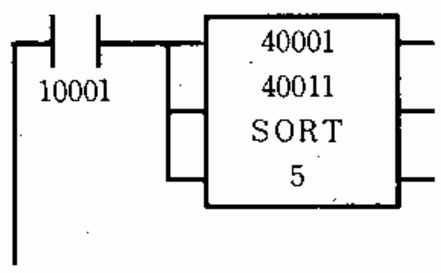
(a) Ladder



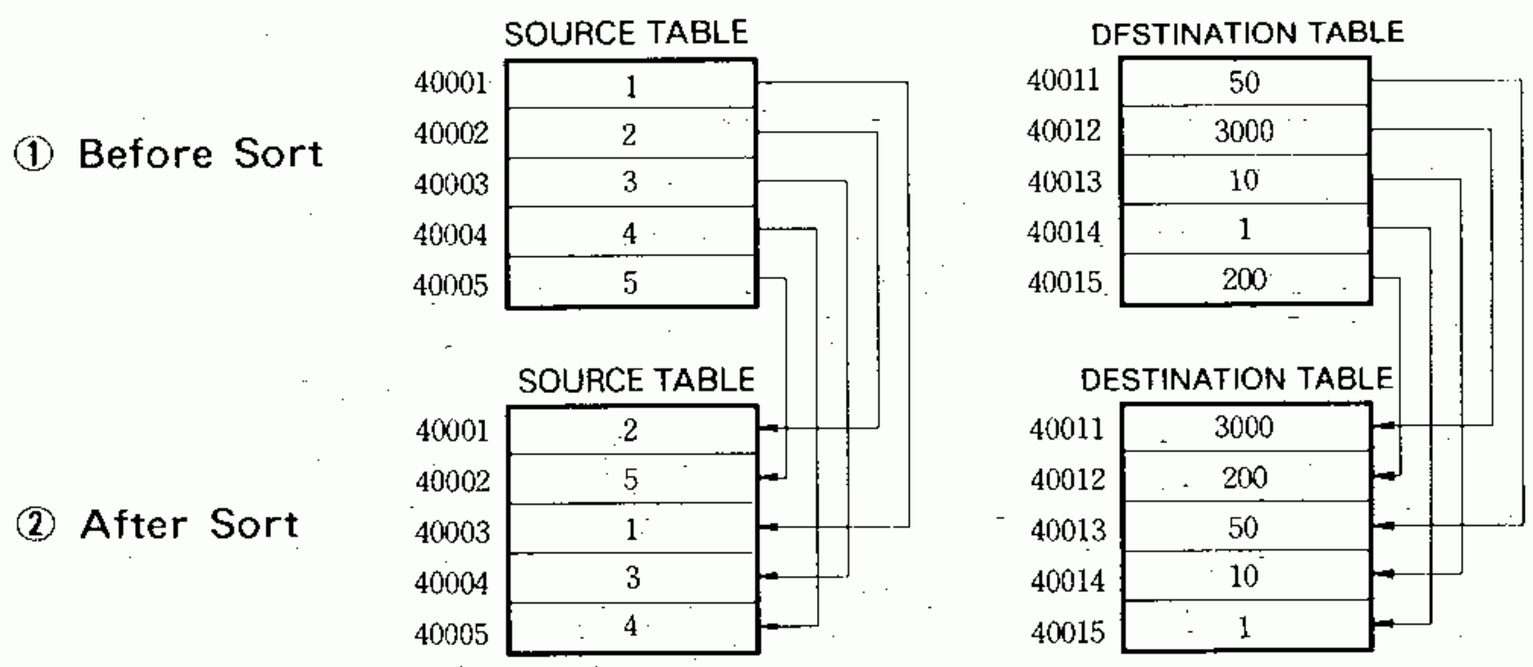
(b) Content of Sort

The SORT shown in (a) executes the SORT which transfers the source table to destination table in the order from the largest value down, because the inputs 1 and 2 will turn ON when input relay 10001 turns ON.

Example 2:



(a) Ladder



(b) Content of Sort

The SORT shown in (a) executes Indexed SORT, because the inputs 1, 2 and 3 turn ON when input relay 10001 turns ON.

5.11.6 Byte Split (BYSL)

(1) Function

It splits the word data of source table into byte data and stores them in destination table.

(2) Form

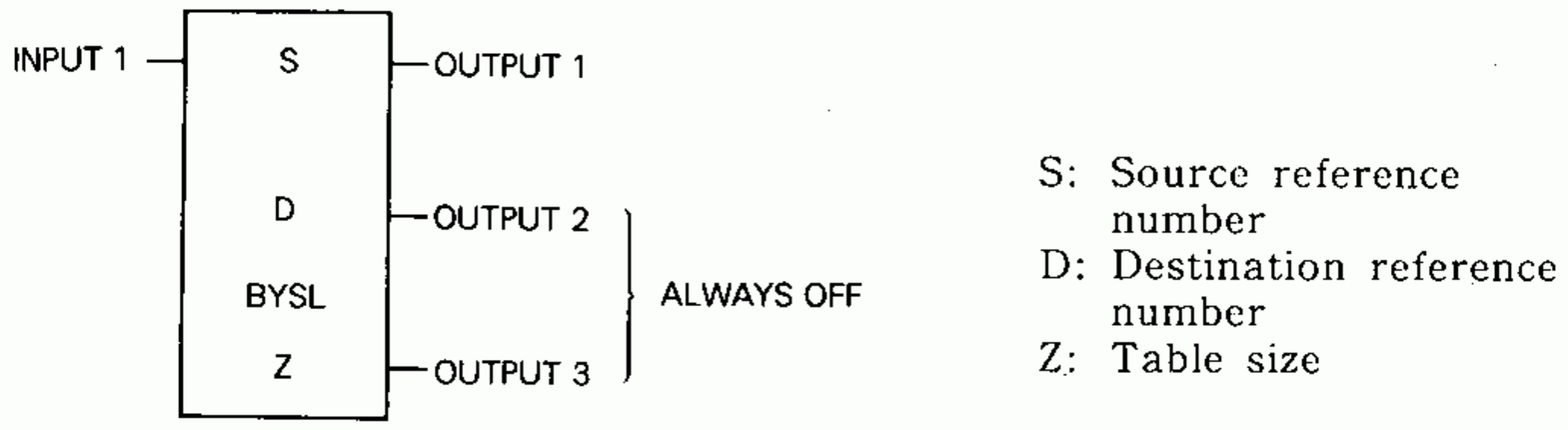


Fig. 5.76 BYSL General Form

- Fig. 5.76 shows the form of byte split.
- BYSL is the symbol denoting byte split.
- BYSL requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.91, specify the needed number for each element.

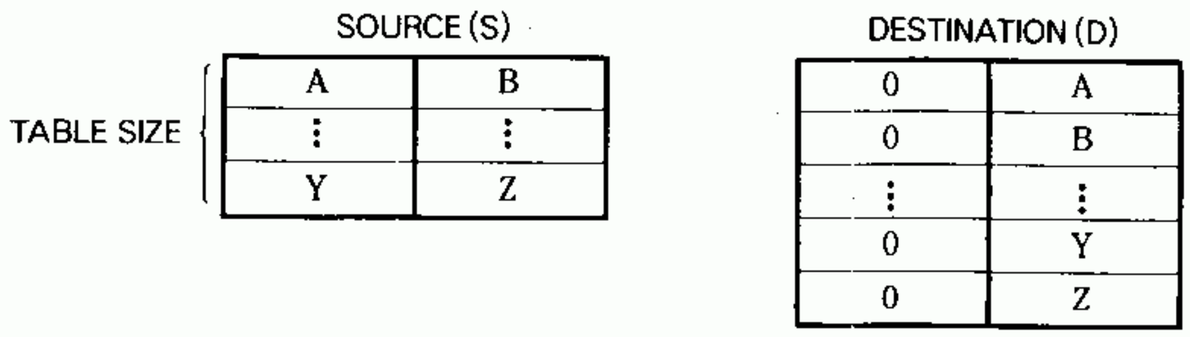
Table 5.91 BYSL Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-100)

(3) Operation

It splits the word data of source table into the high-order byte data and low-order byte data and stores the split data in the destination table. All of the split data are stored in the low-order bytes, and all of the high-order bytes become 0 (zero).

It splits all data in 1 scan cycle and transfers them to the destination table.



5.11.6 Byte Split (BYSL) (Cont'd)

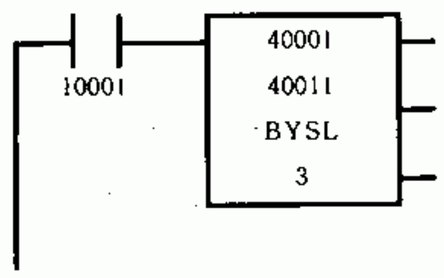
(a) Inputs

- Input 1: Executes byte split and stores the result in the destination table.
- Input 2: } Not used.
- Input 3: }

(b) Outputs

- Output 1: Same as the input 1 ON/OFF status (Copy of the input 1)
- Output 2: } Always OFF.
- Output 3: }

(4) Example



(a) Ladder

SOURCE TABLE

40001	0001	0000	0010	0000
40002	0011	0000	0100	0000
40003	0101	0000	0110	0000

DESTINATION TABLE

40011	ANY PATTERN
40012	
40013	
40014	
40015	
40016	

① Before Byte Split

SOURCE TABLE

40001	0001	0000	0010	0000
40002	0011	0000	0100	0000
40003	0101	0000	0110	0000

DESTINATION TABLE

40011	0000	0000	0001	0000
40012	0000	0000	0010	0000
40013	0000	0000	0011	0000
40014	0000	0000	0100	0000
40015	0000	0000	0101	0000
40016	0000	0000	0110	0000

② After Byte Split

(b) Content of Byte Split

This function is effective for data conversion required when receiving data from ASCII module.

5.11.7 Byte Composition (BYCM)

(1) Function

It composes the byte data of source table into word data and stores them in the destination table. In other words, it converts the data inversely to BYSL in Par. 5.11.6.

(2) Form

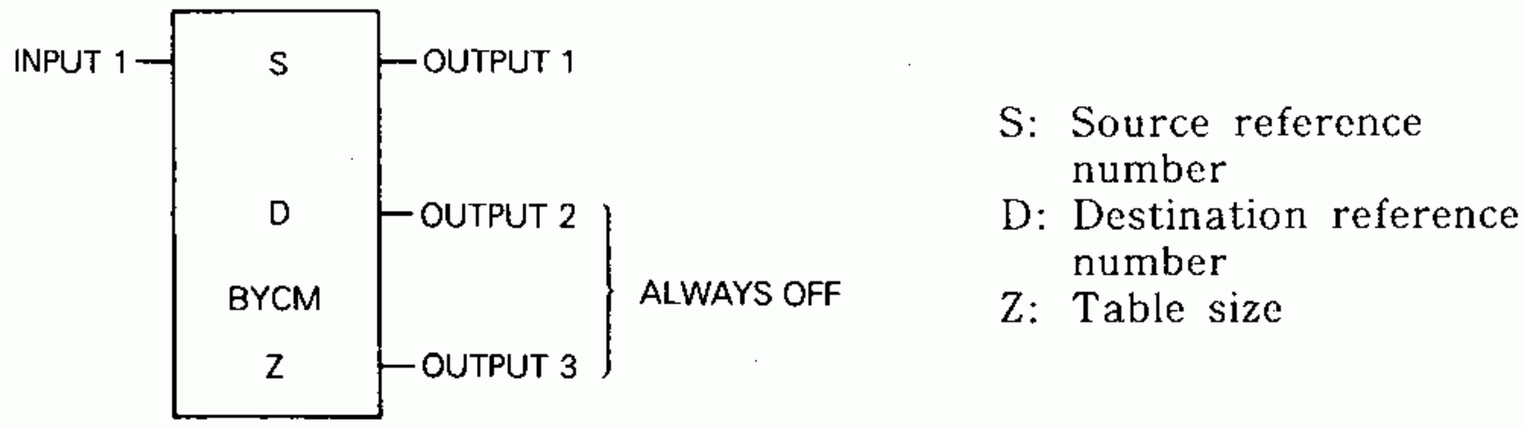


Fig. 5.77 BYCM General Form

- Fig. 5.77 shows the form of byte composition.
- BYCM is the symbol denoting byte composition.
- BYCM requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.92, specify the needed number for each element.

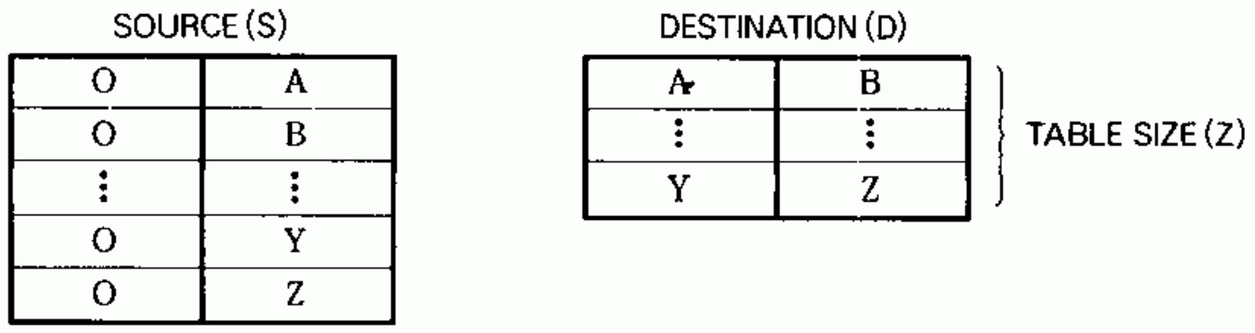
Table 5.92 BYCM Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30127) • Holding register (40001-42047) • Link register (R0001-R1023)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

(3) Operation

It compounds two continuous byte data of source table into one-word data and stores them in destination table. The byte data of odd numbers of source table become the high-order byte, and the byte data of even numbers become the low-order byte.

All of the byte data of source table are composed into word data in 1 scan cycle and transferred to destination table.



5.11.7 Byte Composition (BYCM) (Cont'd)

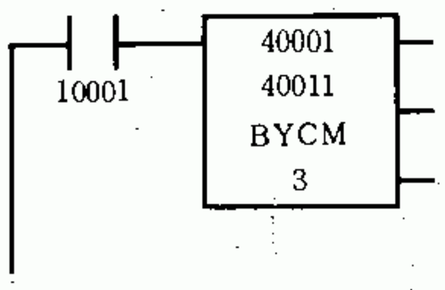
(a) Inputs

Input 1: Executes composition and stores the result in the destination table.
Input 2: } Not used.
Input 3: }

(b) Outputs

Output 1: Same as the Input 1 ON/OFF status.
Output 2: } Always OFF.
Output 3: }

(4) Example



(a) Ladder

SOURCE TABLE

40001	0000 0000 0001 000
40002	0000 0000 0010 0000
40003	0000 0000 0011 0000
40004	0000 0000 0100 0000
40005	0000 0000 0101 0000
40006	0000 0000 0110 0000

DESTINATION TABLE

40011	ANY PATTERN
40012	
40013	

① Before Byte Composition

SOURCE TABLE

40001	0000 0000 0001 0000
40002	0000 0000 0010 0000
40003	0000 0000 0011 0000
40004	0000 0000 0100 0000
40005	0000 0000 0101 0000
40006	0000 0000 0110 0000

DESTINATION TABLE

40011	0001 0000 0100 0000
40012	0011 0000 0100 0000
40013	0101 0000 0110 0000

② After Byte Composition

(b) Content of Byte Composition

BYCM shown in (a) executes conversion (b) when input relay 10001 turns ON.

Even when data other than 0 are in the high-order byte of source table, this function executes the operation described above. And the same operation result is obtained.

5.11.8 Block Addition (BADD)

(1) Function

It adds the content of source table into words or bytes and stores the result in destination table.

(2) Form

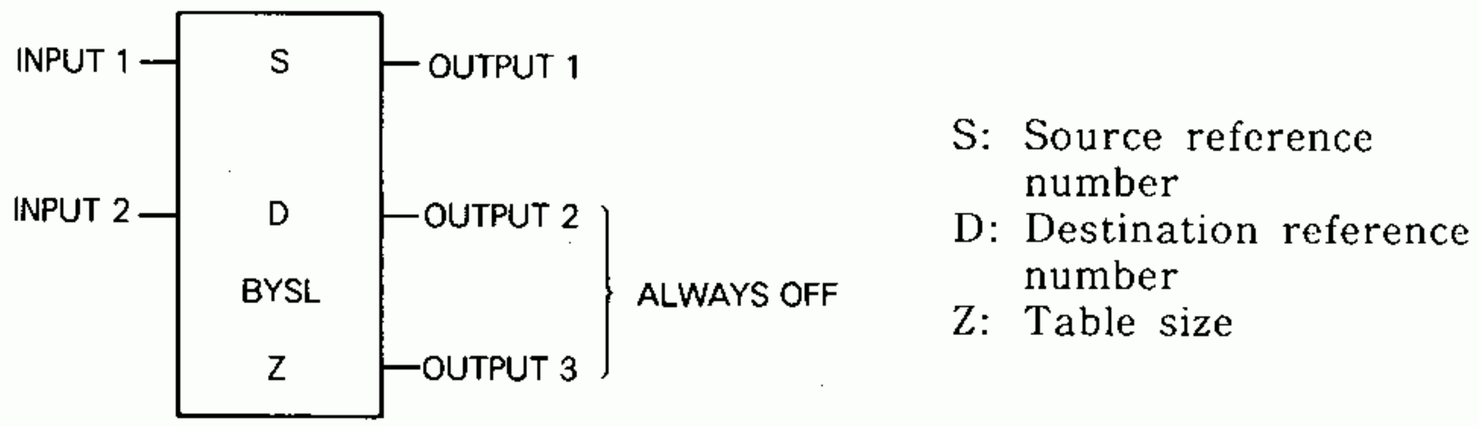


Fig. 5.78 BADD General Form

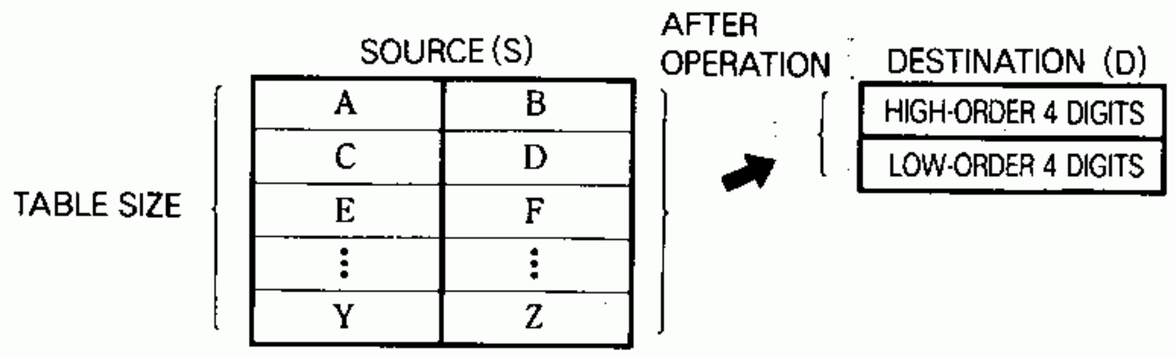
- Fig. 5.78 shows the form of block addition.
- BADD is the symbol denoting block addition.
- BADD requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.93, specify the needed number for each element.

Table 5.93 BADD Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42047) • Link register (R0001-R1023)
Bottom	Table size	• Constant (1-100)

(3) Operation

When the input 1 turns ON, this function adds the content of source table for each word or for each byte according to ON/OFF of the input 2 and stores the result in destination table. It adds up the whole table in 1 scan cycle.



Even when the table size is 100, operation result will not exceed 99,999,999 and operation will be performed correctly. However, operation will not be performed correctly if any negative number is included in the data.

5.11.8 Block Addition (BADD) (Cont'd)

(a) When the input 2 is ON,

Byte: A + B + C + D + • • • + Y + Z

(b) When the input 2 is OFF,

Word: AB + CD + • • • + YZ

(c) The input 3 is not used.

(d) Outputs

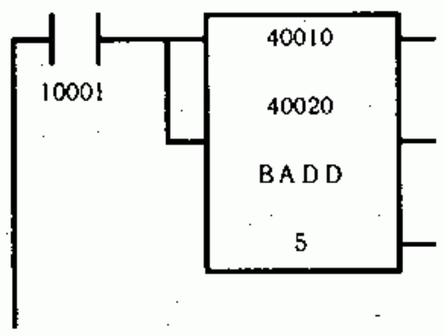
- Output 1: Same as the status of the input 1. (Copy of the input 1)
- Output 2: } Always OFF
- Output 3: }

Table 5.94 shows the operations of BADD.

Table 5.94 Operation of BADD

Input 1	Input 2	Operation	Output 1
ON	ON	Performs addition per each byte.	ON
	OFF	Performs addition per each word.	
OFF	——	Not operated.	OFF

(4) Example



(a) Ladder

SOURCE TABLE

40010	0000	1010	0001	0100
40011	0001	1110	0010	1000
40012	0011	0010	0011	1100
40013	0100	0110	0101	0000
40014	0101	1010	0110	0100

DESTINATION

40020	10
40021	20

① Before Block Addition

SOURCE TABLE

40010	0000	1010	0001	0100
40011	0001	1110	0010	1000
40012	0011	0010	0011	1100
40013	0100	0110	0101	0000
40014	0101	1010	0110	0100

DESTINATON

40020	0
40021	550

② After Block Addition

(b) Content of Block Addition

BADD shown in (a) performs the following operation, because the input 2 turns ON when input relay 10001 turns ON:

10 + 20 + 30 + 40 + 50 + 60 + 70 + 80 + 90 + 100 = 550,

and stores the operation result in the destination table.

5.12 MATRIX

This function group allows matrices to be built in consecutively numbered registers. These matrices are similar to tables previously discussed in that they can be groups of input registers or holding registers depending upon the application requirements. In fact, these same registers can also be operated upon by Move function. Whereas the Move functions operate upon individual registers as elements of tables, the matrix function will operate upon bit patterns within the matrix. Since all registers contain 16 bits, the size of a matrix in bits will be even multiples of 16 (e.g., 32,48,64, 80, etc.). A bit can have one of two states: ON (one) or OFF(zero). Bits within a matrix each have their own identification as illustrated in Fig. 5.79. A matrix of 100 registers contains 1600 bits.

	MSB					LSB
41001	1,	2,	3,	4,	·····	, 16
41002	17,	18,	19,	·····		, 32
41003	33,	34,	·····			, 48
41004	49,	·····				, 64

Fig. 5.79 Sample Matrix Bit Numbering

5.12.1 Types of Matrix

Table 5.95 Types of Matrix

Type	Symbol	Reference Page
Logical AND	AND	201
Logical OR	OR	201
Logical Exclusive OR	XOR	201
Logical Complement	COMP	204
Logical Compare	CMPR	206
Logical Bit Modify	MBIT	210
Logical Bit Sense	SENS	212
Logical Bit Rotate	BROT	216
Logical Multiple Bit Rotate	MROT	219
Logical Byte Rearrangement	TWST	222
Logical Bit Count	BCNT	224

5.12.2 Form of Matrix

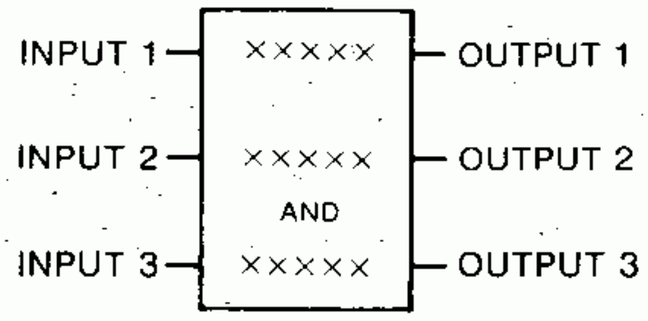


Fig. 5.80 Matrix General Form

The matrix requires three elements placed vertically (top, middle, and bottom), like arithmetic operations and data move. It can be used at any intersection of the 7 lines-by-10 columns matrix (except that the top element cannot be located on lines 6 and 7). AND written between the middle and bottom elements indicates the type of matrix.

Note Byte rearrangement (TWST) alone uses two elements (top and bottom places).

(1) Elements and Their Meanings

The top element is called source and the middle destination. The bottom indicates the size of the matrix table. Each has the same meaning as in the data move.

(2) Reference Number

(a) Numbers specified as reference numbers depend on the type of matrix. See paragraph of each matrix type.

(b) The content of a register (16-bit binary) for the register or the ON/OFF status for discrete signals is operated for the matrix, respectively.

Note When the discrete I/O is specified as a reference number, $n = 16m + 1$ ($m = 0, 1, 2, \ldots$) is required where n is ×××× of 1×××× or 0××××.

(3) Table Size

The constant ×××× is specified as the table size. The range of the fixed value depends on the type of matrix. See paragraph of each matrix type.

Note Specify the number of registers (or of sets of 16 discrete signals) but not the number of bits as the table size.

Except for MBIT, SENS, MROT, and BCNT the source and destination tables have the same size.

(4) Pointer

Only CMPR, MBIT, and SENS functions use a pointer. Its function is the same as that described in Par. 5.9 MOVE.

5.12.3 AND, OR, XOR

Each logical function is provided with a similar function block , so these functions are described in this section.

(1) Function

These functions perform logical operations including AND, OR and XOR between source table and destination table and stores the results in the destination table.

(2) Form

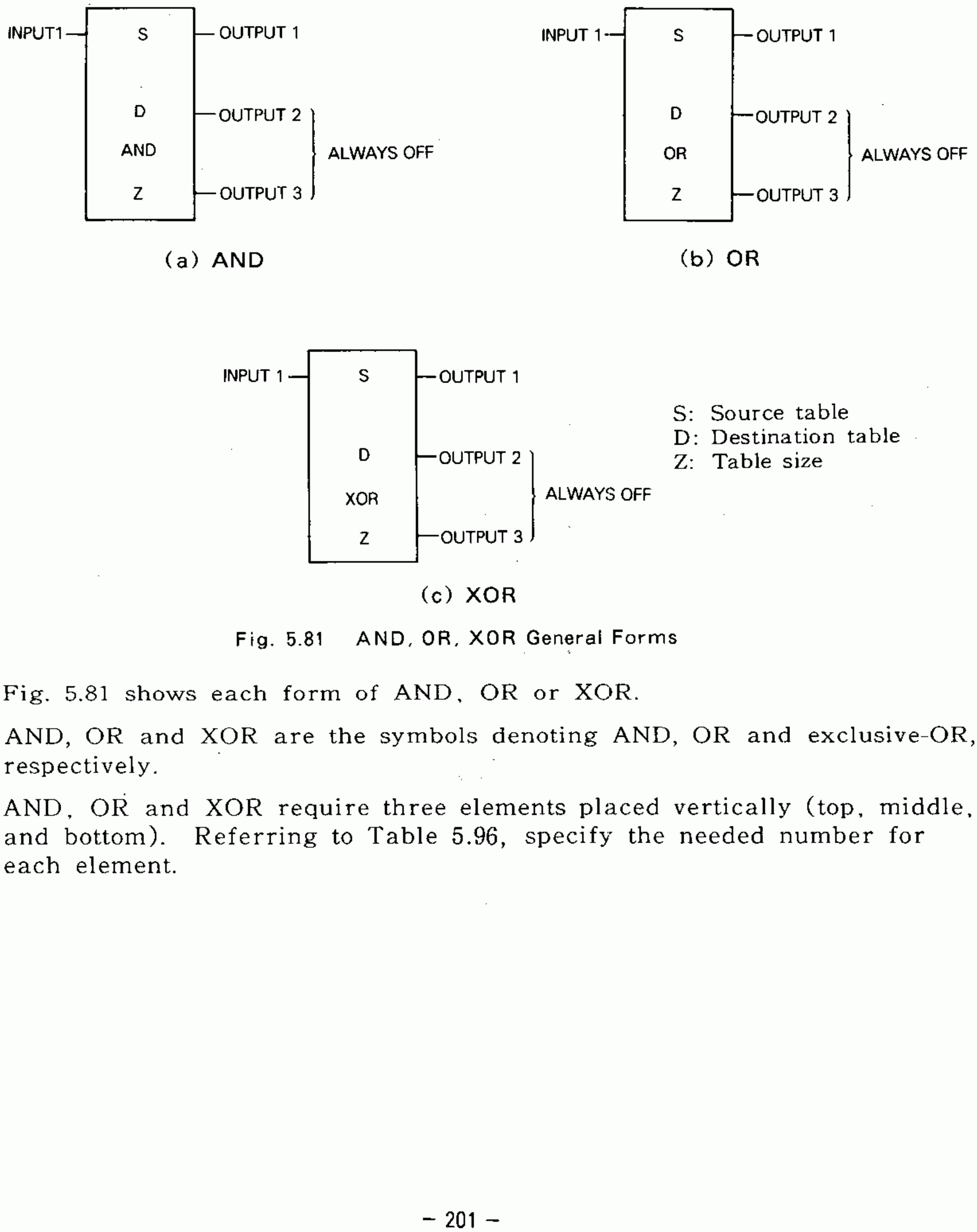


Fig. 5.81 AND, OR, XOR General Forms

- Fig. 5.81 shows each form of AND, OR or XOR.
- AND, OR and XOR are the symbols denoting AND, OR and exclusive-OR, respectively.
- AND, OR and XOR require three elements placed vertically (top, middle, and bottom). Referring to Table 5.96, specify the needed number for each element.

Table 5.96 AND, OR, XOR Elements

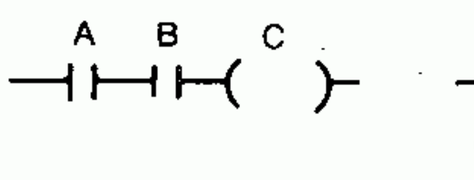
Element	Description	Specified Number
Top	Source reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Coil (00001-02033) • Link coil (D0001-D1009) • Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	The followings depend on number specified in top or middle. • Coil (1-100) • Input relay (1-32) • Link coil (1-64) • Various registers (1-100)

(3) Operation

Fig. 5.82 shows the truth tables and equivalent relay circuits of AND, OR, and XOR applied to 1 bit signal.

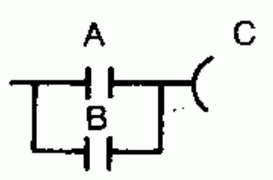
AND (A · B = C)

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



OR (A + B = C)

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1



XOR (A ⊕ B = C)

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

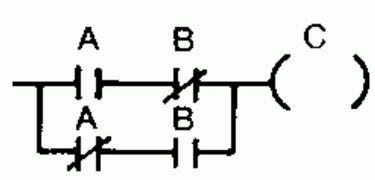
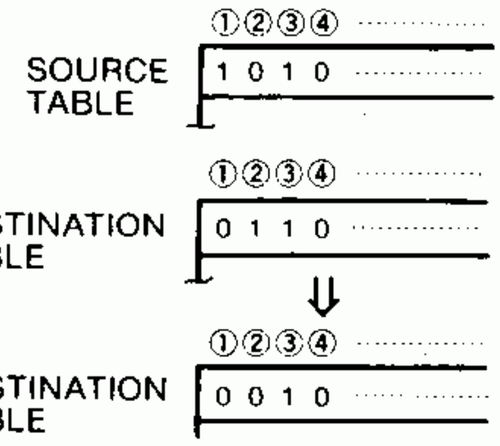


Fig. 5.82 Truth Tables and Equivalent Relay Circuits

This function causes two matrices of equal length to be logically AND'ed, OR'ed or XOR'ed together and the results stored for reference by any other logic function. The result of the logical AND will be a one (ON) bit only if both bits (one from each matrix) are one bits; otherwise, the result will be zero (OFF) bit. One bit from each matrix with the same identification number will be combined in accordance to the truth table.

Example: AND



Note The bit numbers are circled. AND is applied to the bit pair having the same bit number and result is stored in the bit having the same number.

All bits in the matrices will be operated upon every scan the function is enabled. The Source matrix is not altered only copied. The current content of the Destination matrix is used for the AND, OR, or XOR operation and then replaced with the result of the operation. The Destination matrix is altered by the AND, OR, or XOR operation. The Matrix AND operation is useful to clear large groups of registers to zero (when ANDed with a matrix of zeros) or to construct masks within the controller. The Matrix OR operation is useful to construct masks within the controller. The Matrix XOR operation is useful to detect differences between two matrices within one scan and, when operated upon the same matrix in both Source and Destination, to clear a matrix to zero. See Example 2.

(a) Inputs

Only the input 1 is used with the these functions. When this input node receives power flow, the each operation is performed. Every scan, when enabled, the AND, OR or XOR will operate upon the entire content of the matrices. Transitional contact can be used if a single operation is desired.

(b) Outputs

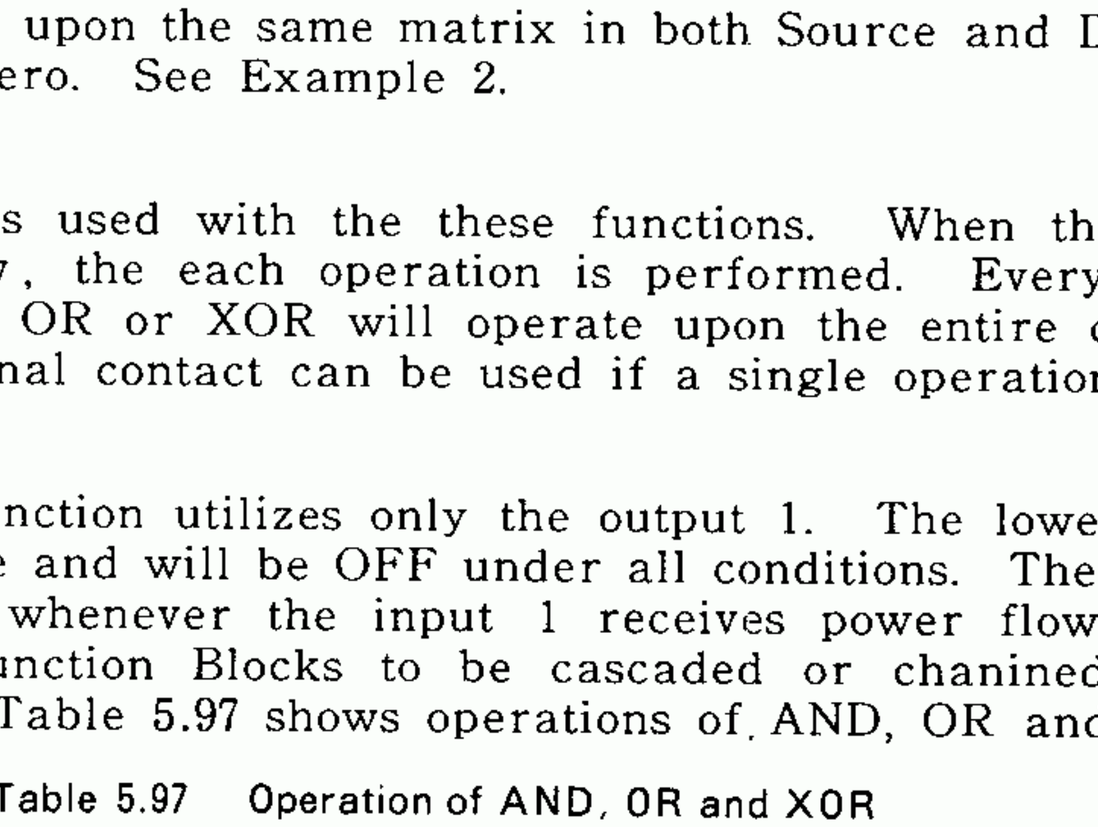
The each matrix function utilizes only the output 1. The lower two outputs have no significance and will be OFF under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. Thus, the output 1 allows Function Blocks to be cascaded or chanined horizontally within a network. Table 5.97 shows operations of AND, OR and XOR.

Table 5.97 Operation of AND, OR and XOR

Input 1	Operations	Output 1
ON	AND, OR, XOR operated.	ON
OFF	Not operated.	OFF

(4) Example

Example 1:



Example 2:

This XOR is used to clear the source and destination tables when operated upon the same matrix in both Source and Destination.

5.12.4 Complement (COMP)

(1) Function

This function causes the content of one matrix to be complemented (all ones replaced by zeros, and zeros by ones) and placed in another matrix for reference by any other function. The entire matrix is operated upon every scan the function is enabled by power flow to the input. The result of the Complement operation is placed in the Destination matrix; the previous content of this matrix is lost. The Source matrix is not altered only copied.

The Matrix Complement is useful to alter normally closed inputs to the same base as normally open inputs or to move masks within the controller.

(2) Form

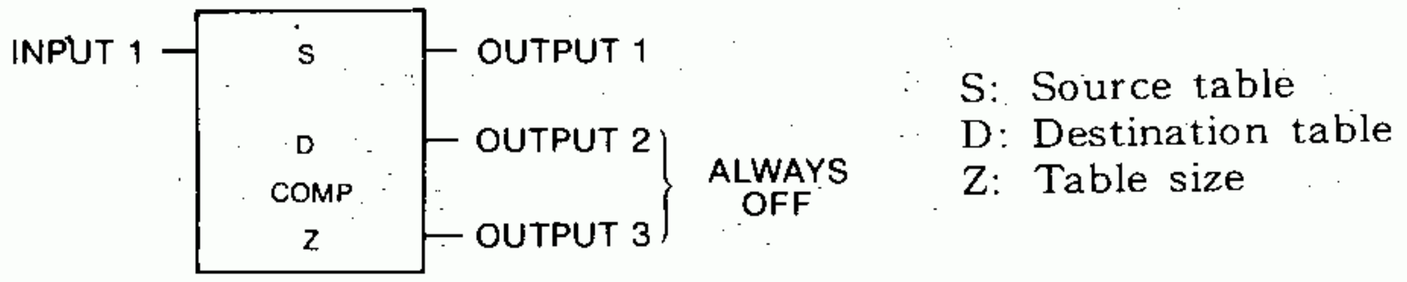


Fig. 5.83 COMP General Form

- Fig. 5.83 shows the form of complement.
- COMP is the symbol denoting complement.
- COMP requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.98, specify the needed number for each element.

Table 5.98 COMP Elements

Element	Description	Specified Number
Top	Source reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Coil (00001-02033) • Link coil (D0001-D1009) • Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	The followings depend on number specified in top or middle. • Coil (1-100) • Input relay (1-32) • Link coil (1-32) • Various registers (1-100)

(3) Operation

(a) Inputs

Only the input 1 is used with the COMPLEMENT function. When this input node receives power flow, the COMPLEMENT operation is performed. Every scan , when enabled , the COMPLEMENT will operate upon the entire content of both matrices. Transitional contacts can be used if a single operation is desired.

(b) Outputs

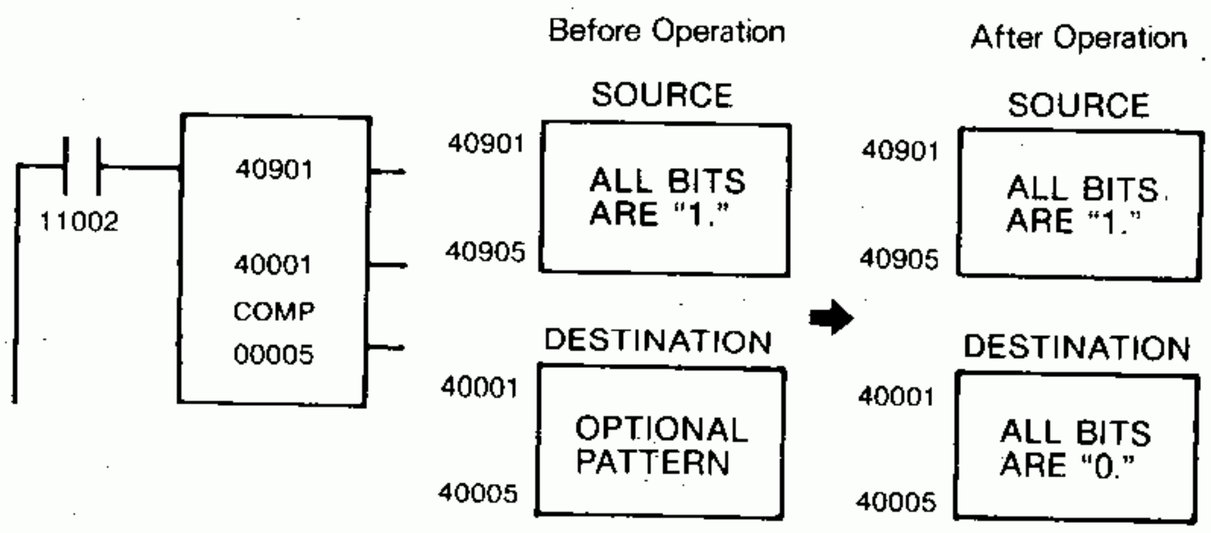
The COMPLEMENT matrix function utilizes only the output 1. The lower two outputs have no significance and will be OFF under all conditions.
The output 1 will supply power flow whenever the input 1 recives power flow. Thus the output 1 allows function Blocks to be cascaded or chanined horizontally with a network.

Table 5.99 shows operation of COMP.

Table 5.99 Operation of COMP

Input 1	Operations	Output 1
ON	Complemented	ON
OFF	Not complemented.	OFF

(4) Example



5.12.5 Compare (CMPR)

(1) Function

This function causes two matrices to be compared on a bit-by-bit basis; their contents are not altered only examined. When enabled, the compare function will examine one bit from each matrix with the same identification number. If these bits agree (both zero or both ones) the next bit from each matrix is compared; however, if they do not agree, the compare function will halt.

(2) Form

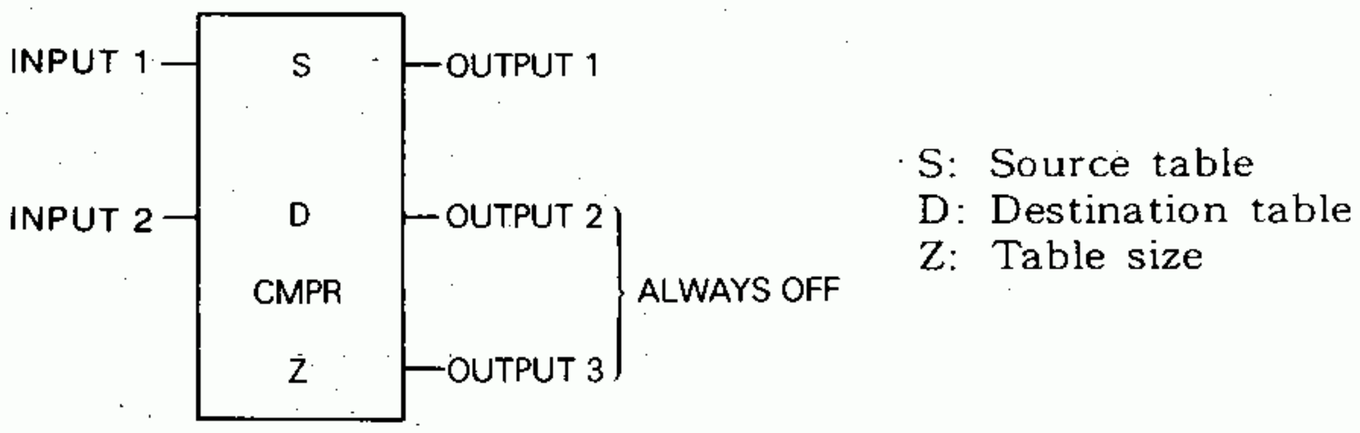


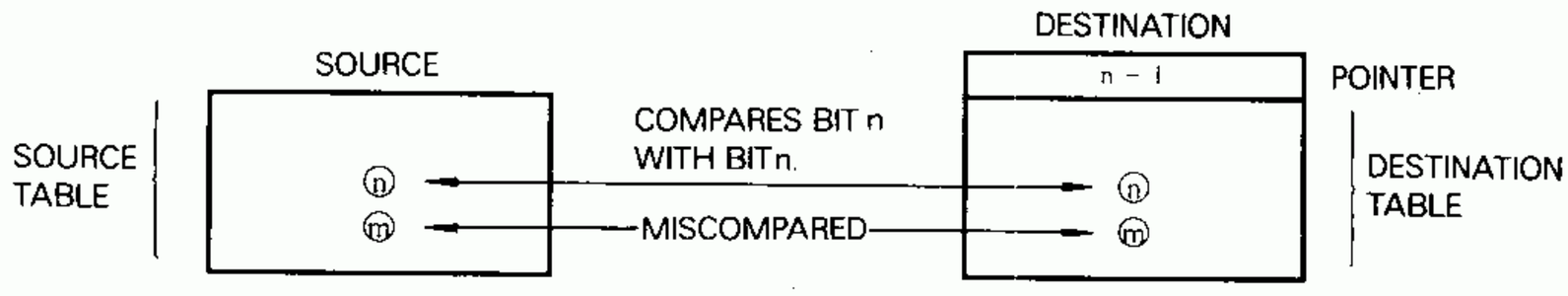
Fig. 5.84 CMPR General Form

- Fig. 5.84 shows the form of compare.
- CMPR is the symbol denoting compare.
- CMPR requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.100, specify the needed number for each element.

Table 5.100 CMPR Elements

Element	Description	Specified Number
Top	Source reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	The followings depend on number specified in top or middle. • Coil (1-100) • Input relay (1-32) • Link coil (1-64) • Various registers (1-100)

(3) Operation



Each scan the compare is performed, either the end of the matrix is located or a miscompare is encountered. If both matrices are identical, the entire length, up to 100 registers or 1600 bits (such as discrete inputs/outputs), is compared each and every scan.

An output is used to indicate the result of compare. It will be ON if a miscompare is detected and OFF if the end of the matrices is reached. A pointer is the only register whose content is altered by the Compare. This pointer is referred to by the Destination block and indicates which specific bit was responsible for the miscompare, if the operation is terminated by a miscompare (output ON). The pointer can also be used to cause the comparison to begin on bits other than at the beginning; the compare function always proceeds towards the end (high bit number) of the matrices. Before the Compare, the pointer will be incremented; if the pointer is at the end of the matrix or longer, it will be reset to one prior to beginning the compare operation.

Note
1. The contents of the pointer can be changed by another logic circuit. To start compare at bit i, set i-1 to the pointer before starting.
2. If there are no miscompares, the pointer will be at the end of the matrices when the comparison is completed.

(a) Inputs

Only the upper two inputs are used with the COMPARE function. Every scan the input 1 receives power flow, the Compare operation is performed. Up to 1600 pairs of bits can be compared each scan the input is enabled. Transitional contacts can be used if a single operation is desired.

The input 2 controls the reset of the pointer. When this input receives power flow, the pointer will be reset to zero prior to the comparison; otherwise, the comparison begins at the current value of the pointer plus one. If the pointer is at the end of the matrix or greater, it will be reset to zero prior to the comparison regardless of the state of this input. Matrices begin at bit number one (not zero), thus resetting implies the value one.

(b) Outputs

The COMPARE matrix function utilizes all three available outputs. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Block to be cascaded or chanined horizontally within a network. The output 2 will supply power flow if the comparison has detected a miscompare; this output will be OFF if either no comapre is being performed (top input does not receive power flow), or if no miscompares have been detected and the end of the matrices have been reached. The output 3 senses the Source matrix bit if and only if a miscompare has been detected; this output will supply power flow when this bit is a one and be OFF if the bit is a zero. Using the lower two outputs, logic can be built to determine the exact bit configuration causing a miscompare (Source = one and Destination = zero, or vice versa).

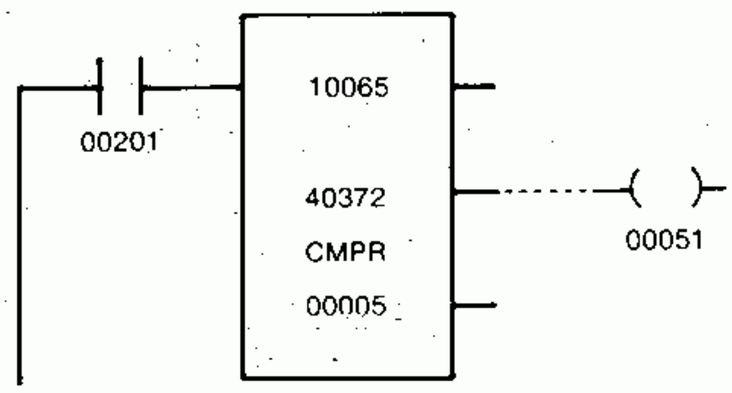
5.12.5 Compare (CMPR) (Cont'd)

Table 5.101 shows operation of CMPR.

Table 5.101 Shows Operation of CMPR

Input 1	Input 2	Operations	Miscompare	Condition	Output 1	Output 2	Output 3
ON	OFF	Checks from the bit No. next to pointer.	No	——	ON	OFF	OFF
			Yes	S = 1 D = 0		ON	ON
				——			OFF
	ON	Turns pointer to 0 and checks from the bit No. at the head of table.	No	——		OFF	OFF
			Yes	S = 1 D = 0		ON	ON
				——			OFF
OFF	OFF	Not operated.	——	——	OFF	OFF	OFF
	ON	Turns pointer to 0. Not operated.					

(4) Example



(a) Ladder

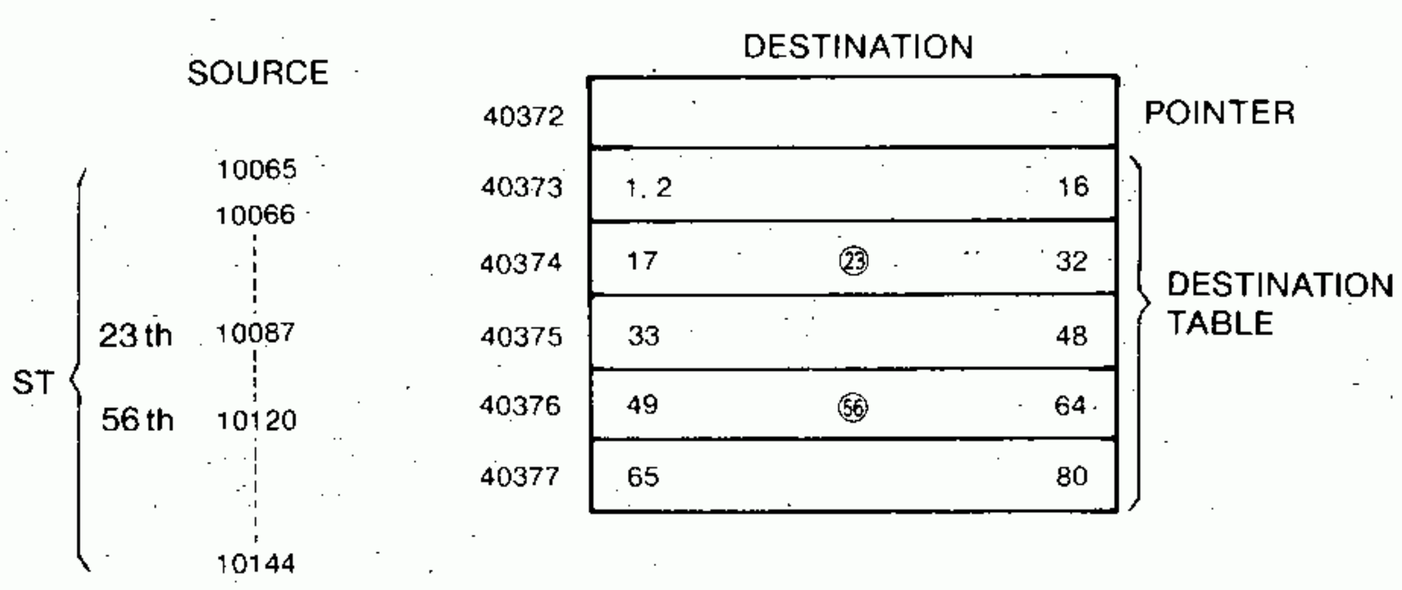


Figure above illustrates a typical COMPARE Matrix function. If the pointer (register 40372) contains the value zero or one with input 10056 is energized, the comparison begins at input 10056 and bit 1. The entire matrix all 80 bits will be compared to the inputs unless a miscompare is detected. However, if bit 23 in register 40372 is energized. If input 10087 is ON and bit 23 in the destination matrix (40373-40377) is a zero, coil 00051 will be energized.

On the next scan with input 00201 still energized and the content of register 40372 not altered, the comparison will start at bit 24 and proceed towards the end of the matrix. If input 10120 is OFF and bit 56 set to a one value in the matrix 40373-40377, an additional miscompare is detected. The value in register 40372 is now 0056 and coil 00051 remains ON, 00051 since the Source bit is a zero (input 10120 OFF). The pointer value of the first miscompare is lost and replaced by the location of the second miscompare. To retain miscompare locations, they should be saved in another location, such as a table, with a register to table move function. This move function can be controlled by the coil 00051 in this example.

On the next scan, unless there are additional miscompares, the comparison begins at bit 57 and reaches the end of the matrix. The pointer will contain the value 81, coils 00051 will be OFF. On the next (fourth) scan, the comparison begins again and will detect bit 23 as a miscompare again unless the input or bit status is altered to prevent the miscompare. The compare funciton is very powerful and very fast. Unless action is taken, pointer values will be replaced by other values and repetitive miscompares detected. At best, all bits are compared every scan when they all agree, and at worst case one bit is compared each scan when they all disagree.

5.12.6 Modify (MBIT)

(1) Function

This function allows individual bits in a matrix to be altered. Only one bit per scan can be affected by this function; all other bits retain their state. Bits can be either set to a one (ON) condition or cleared to a zero (OFF) condition. A pointer is used to indicate which bit is to be modified.

(2) Form

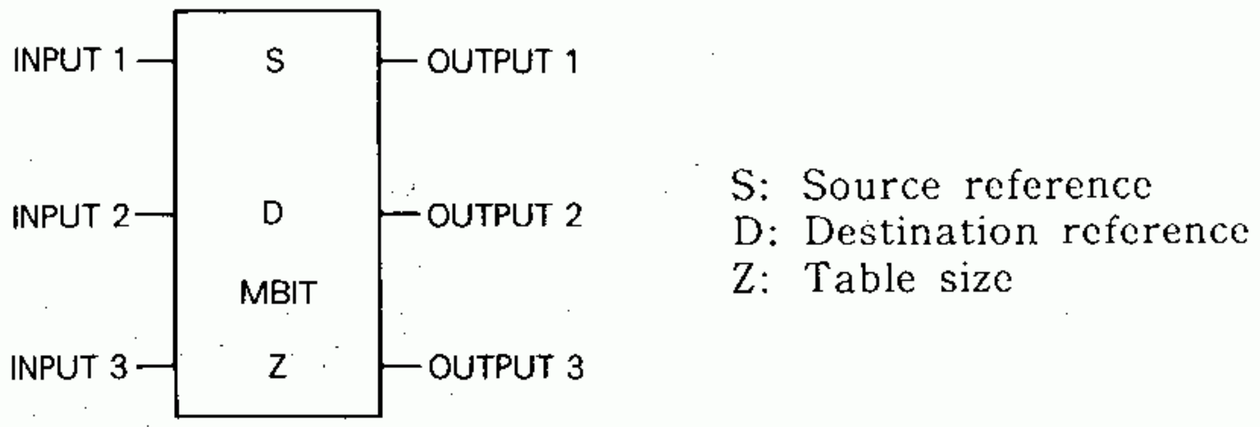


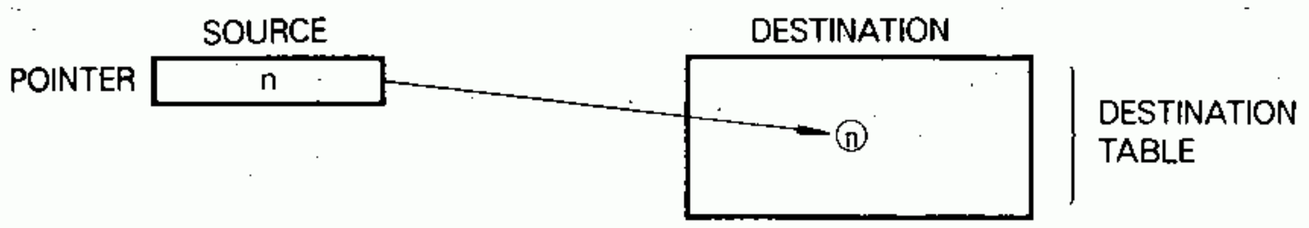
Fig. 5.85 MBIT General Form

- Fig. 5.85 shows the form of modify.
- MBIT is the symbol denoting modify.
- MBIT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.102, specify the needed number for each element.

Table 5.102 MBIT Elements

Element	Description	Specified Number
Top	Source reference number	• Constant K (0001-9600) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Coil (00001-02033) • Link coil (D0001-D1009) • Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• For coil specified in middle : Constant K (1-128) • For link coil specified in middle : Constant K (1-64) • For others specified : Constant K (1-600)

(3) Operation



Note Set bit n to "1" or "0".

(a) Inputs

All three inputs are used with the Bit Modify function. Every scan the input 1 receives power flow, a bit is altered. Any of up to 9600 bits can be modified by this function. Transitional contacts can be used if a single operation is desired. The input 2 controls how that bit is to be modified. When this input receives power flow, the bit will be set to a one (ON) condition; no power flow results in the bit being cleared to a zero (OFF) condition. The current status (ON/OFF) of the bit has no effect on the result after this function. The input 3 when receiving power flow will cause the pointer, if stored in register only, to be incremented after the bit is altered. The pointer is not incremented if it is a constant, stored in an input register, or there is no power flow to the input 3. The pointer, if incremented beyond the size of the matrix, will be reset to one automatically.

(b) Outputs

The Bit Modify matrix function utilizes all three outputs. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function blocks to be cascaded or chanined horizontally within a network.

The output 2 supplies power flow whenever the bit is ON, one, after the operation is performed. Thus this output can be described as sensing the resultant bit or merely copying the state of the input 2 with a successful function. The output 3 will supply power flow if the pointer's magnitude is beyond the size of the matrix (pointer too large). Thus if the pointer in an input register or a holding register without automatic incrementing exceeds the matrix size, no operation is performed and this output is used to indicate the error condition.

MBIT function and status at I/O ON are shown in Tables 5.103 and 5.104, respectively.

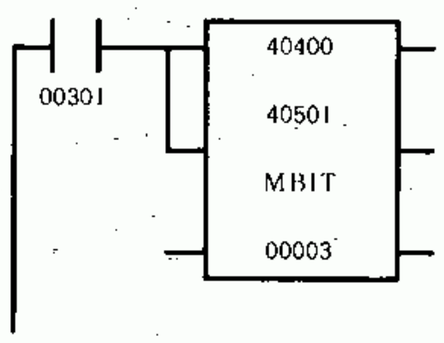
Table 5.103 MBIT Functions

Input ×	Functions	Remarks
Input 1	Sets or clears bits.	These actions are not performed, if the content of pointer is either 0 or exceeds the maximum bit number.
Input 2	Sets "1" at ON; clears "1" (sets to "0") at OFF.	Effective only when the input 1 is ON.
Input 3	Only when register is used as source, it adds + 1 to register content, after execution.	Effective only when the input 1 is ON.

Table 5.104 MBIT Output Status

Output ×	Status
Output 1	Same as the state of ON/OFF of the input 1 (Copy of input 1)
Output 2	Turns ON, when bit is "1" after execution.
Output 3	Turns ON, when bit No. exceeds the maximum bit No. determined according to table size. (regardless of ON/OFF state of the input 1)

(4) Example



(a) Ladder Circuit

① Before Modify

② After Modify

(b) Content of Modify

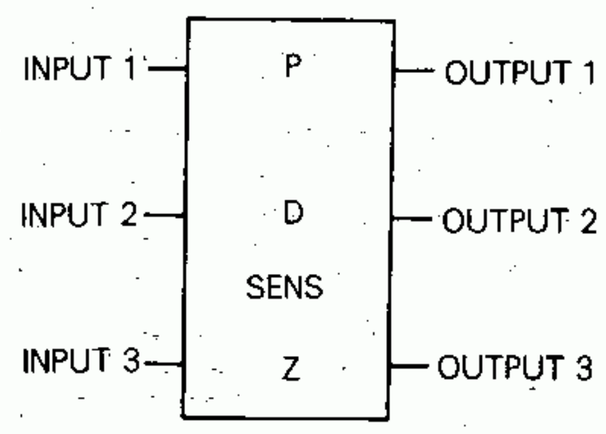
Figure above illustrates a typical Bit Modify Matrix function. If the pointer (register 40400) contains the value three, when coil 00301 is energized, the bit in the matrix 40501-40503 at location three will be set to a one. The control that sets the bit (in lieu of clearing the zero) is the vertical connection to the middle input. As long as coil 00301 is energized, bit three will be set every scan; unless other logic clears this bit, no change in the bit will be detectable.

5.12.7 Sense (SENS)

(1) Function

This function allows individual bits in a matrix to be examined, but not altered. An output is used to indicate one (ON) bits with power flow and a zero (OFF) bits without power flow. The operatoin of this function is similar to the Bit Modify previously discussed, except that no bits are modified. The status of only one bit can be obtained per scan.

(2) Form



P: Pointer
D: Destination reference number
Z: Table size

Fig. 5.86 SENS General Form

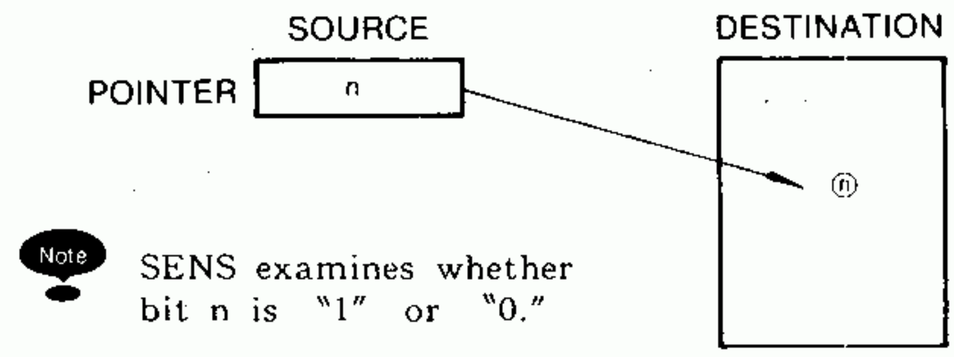
- Fig. 5.86 shows the form of sense.
- SENS is the symbol denoting sence.
- SENS requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.105, specify the needed number for each element.

Table 5.105 SENS Elements

Element	Description	Specified Number
Top	Pointer	• Constant K (0001-9600) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• For coil specified in middle : Constant (1-128) • For input relay specified in middle : Constant (1-32) • For link coil specified in middle : Constant (1-64) • For register specified in middle : Constant (1-600)

(3) Operation

This function allows individual bits in a matrix to be examined, but not altered. An output is used to indicate one (ON) bits with power flow and a zero (OFF) bits without power flow. The operation of this function is similar to the Bit Modify previously discussed, except that no bits are modified. The status of only one bit can be obtained per scan.



(a) Inputs

All three inputs are used with the Bit Sense function. Every scan the input 1 receives power flow, a bit is located in a matrix and its status is obtained. Any of up to 9600 bits can be obtained with this function. Transitional contacts can be used if a single operation is desired.

The input 2 controls the incrementing of the pointer. When this input receives power flow, the pointer, if stored in a holding register only, will be incremented after the bit is examined. Both the inputs 1 and 2 must receive power flow for the pointer to be inremented. The pointer is not incremented if it is a constant, stored in an input register, or there is no power flow to the input 2. The pointer, if incremented beyond the size of the matrix, will be reset to one automatically. The pointer is also reset to one if the input 1 receives power flow; this resetting is accomplished prior to sensing the bit as required by the top input.

(b) Outputs

The bit Sense matrix function utilizes all three outputs. The output 1 will supply power flow whenever the input 1 receives power flow. Thus the output 1 allows Function Blocks to be cascaded or chained horizontally within a network. The output 2 supplies power flow whenever the bit being sensed (addressed by the pointer) is a one (ON) bit; this output will not supply power flow whenever the bit is a zero (OFF) bit. The input 2 is the resultant of the sense function. The output 3 will supply power flow if the pointer's magnitude is beyond the size of the matrix (pointer too large). Thus if the pointer is a holding register with automatic pointer incrementing and it exceeds the matrix size, no operation is performed and this output is used to indicate the error condition.

SENS function and status at I/O ON are shown in Tables 5.106 and 5.107, respectively.

Table 5.106 SENS Functions

Input ×	Functions	Remarks
Input 1	Executes SENS.	It will not execute SENS, if the pointer content is 0 or if it exceeds the maximum bit No. determined according to table size.
Input 2	Only when the pointer is a holding register or a link register, it adds + 1 to the content of the register after execution of SENS.	The input 1 must be ON.
Input 3	Only when the pointer is a holding register or a link register, it turns the register content to 1.	Regardless of the input 1 ON and OFF

Table 5.107 SENS Output Status

Output ×	Status
Output 1	Same as the input 1 ON/OFF status.
Output 2	Turns ON, when the specified bit No. is "1".
Output 3	Turns ON, when the specified bit No. exceeds the maximum bit No. determined according to table size. (Regardless of the input 1 ON and OFF)

(4) Example

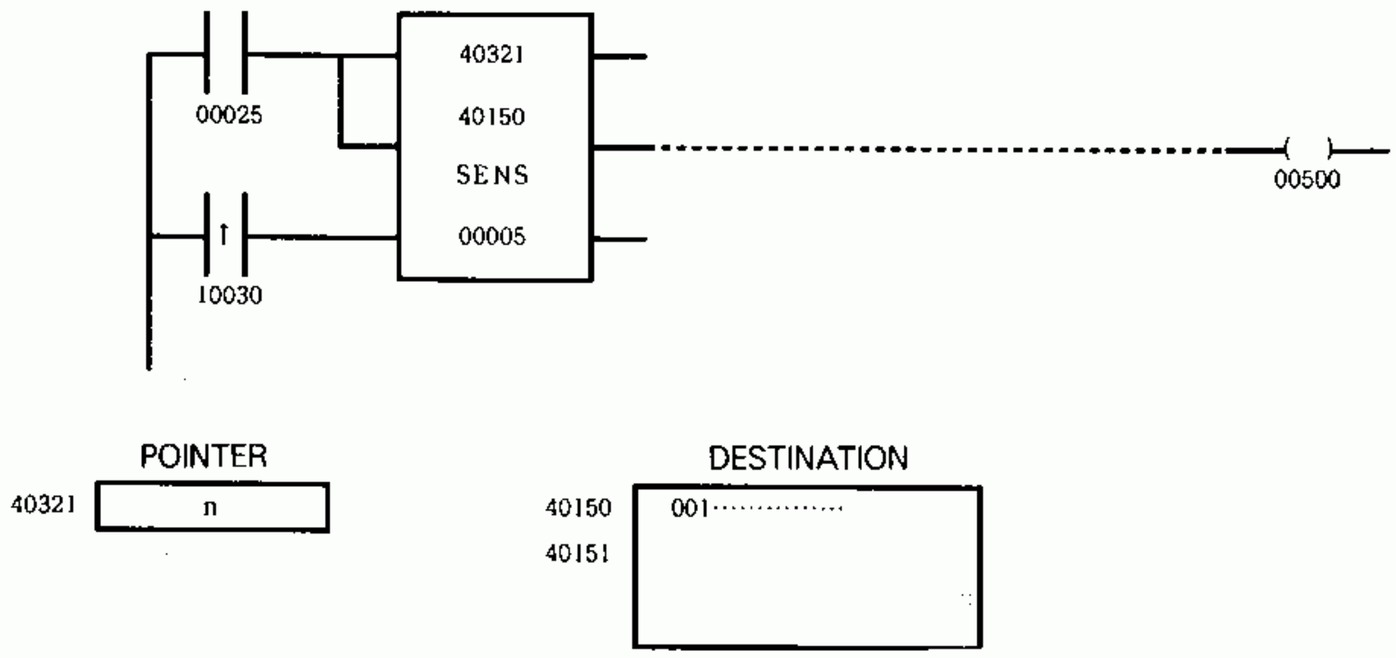


Figure above illustrates a typical Bit Sense Matrix function. If the pointer (register 40321) contains the value zero when input 00025 is energized, coil 00500 will be off during that first scan. Since there is a vertical connection for the power flow to input 2 the pointer will increment by one each scan input 00025 is energized.

Thus on the second scan that this input is ON, the pointer will be at the value one, and the first bit in the matrix (registers 40321-40324) will be sensed. This bit is a zero, and coil 00500 remains OFF; this coil will also be OFF for the third scan while bit two is sensed. When the pointer is incremented to the value three, coil 00500 will be ON sensing bit three as a one bit.

As long as input 00025 is energized, the sense function "walks" through the matrix at the rate of one bit per scan; coil 00500 indicates the status of each bit. The sensing will return to bit one after bit 80 automatically if input 00025 is energized for a sufficiently long period. Whenever input 10030 is energized, the sensing will return to bit one regardless of the pointer's value; since a transitional contact is used, input 1003 must be deenergized and then re-energized to affect the pointer.

5.12.8 Rotate (BROT)

(1) Function

BROT shifts all bits of the source table, one bit to the left or right, and stores the result in the destination table all in a scanning cycle.

(2) Form

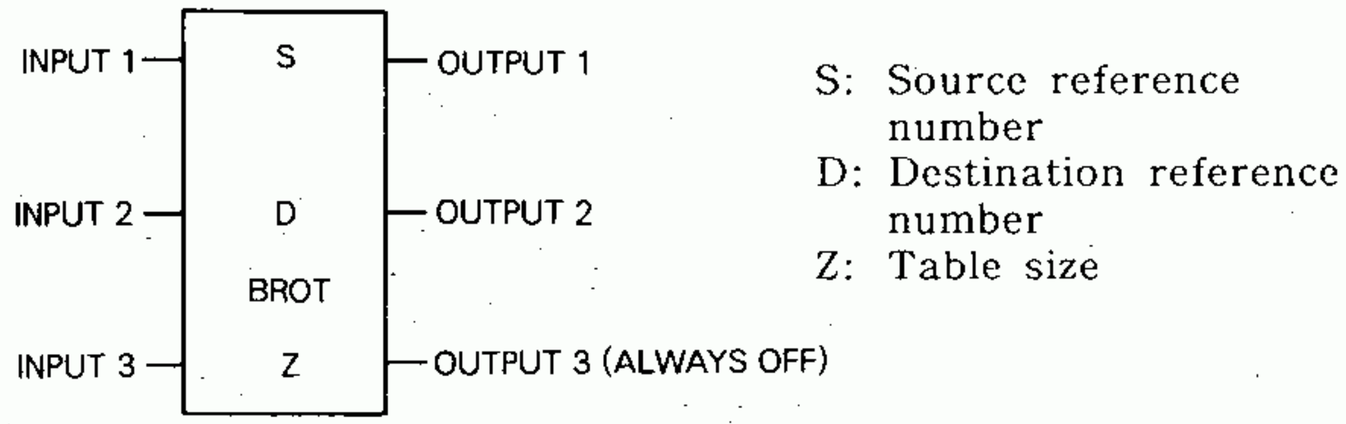


Fig. 5.87 BROT General Form

- Fig. 5.87 shows the form of rotate.
- BROT is the symbol denoting rotate.
- BROT requires three element placed vertically (top, middle, and bottom). Referring to Table 5.108, specify the needed number for each element.

Table 5.108 BROT Elements

Element	Description	Specified Number
Top	Source reference number	• Coil (00001-02033) • Input relay (10001-10497) • Link coil (D0001-D1009) • Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Coil (00001-02033) • Link coil (D0001-D1009) • Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• For link coil specified in top and middle : Constant (1-64) • For coil specified in top and middle : Constant (1-128) • For input relay specified in top and middle : Constant (1-32) • For others : Constant (1-100)

(3) Operation

(a) Inputs

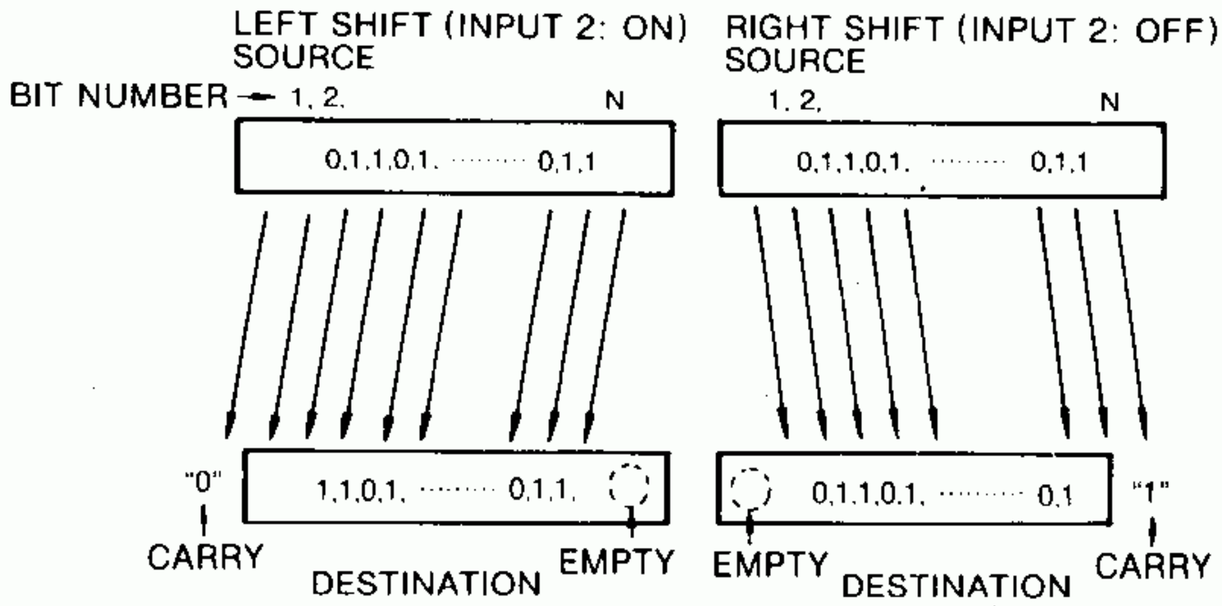
All three inputs are used with the Bit Rotate function. Every scan the input 1 receives power flow, all bits in the matrix will be rotated one position. Up to 1600 bits are moved with this function in one scan.

The input 2 controls the direction of the rotation. If this input receives power flow, the matrix will be rotated towards the left (bit 17 into 16, bit 16 into 15,.... bit 3 into 2, bit 2 into 1, and bit 1 rotated out of the matrix). If the input 2 does not receive power flow, the matrix will be rotated towards the right (bit 1 into 2, bit 2 into 3, ... bit 15 into 16, bit 16 into 17, etc.); the last bit will be rotated out of the matrix. The input 3 controls what happens to the single bit location vacated by the rotate to create either a shift operation or a true rotate. When this input does not receive power flow, the bit rotated out is ignored and vacant location at the opposite end of the matrix will be filled with zero; this is a shift operation. If this input receives power flow, the bit rotated out is carried around unchanged and entered into the opposite end of the matrix; this is a true rotate operation.

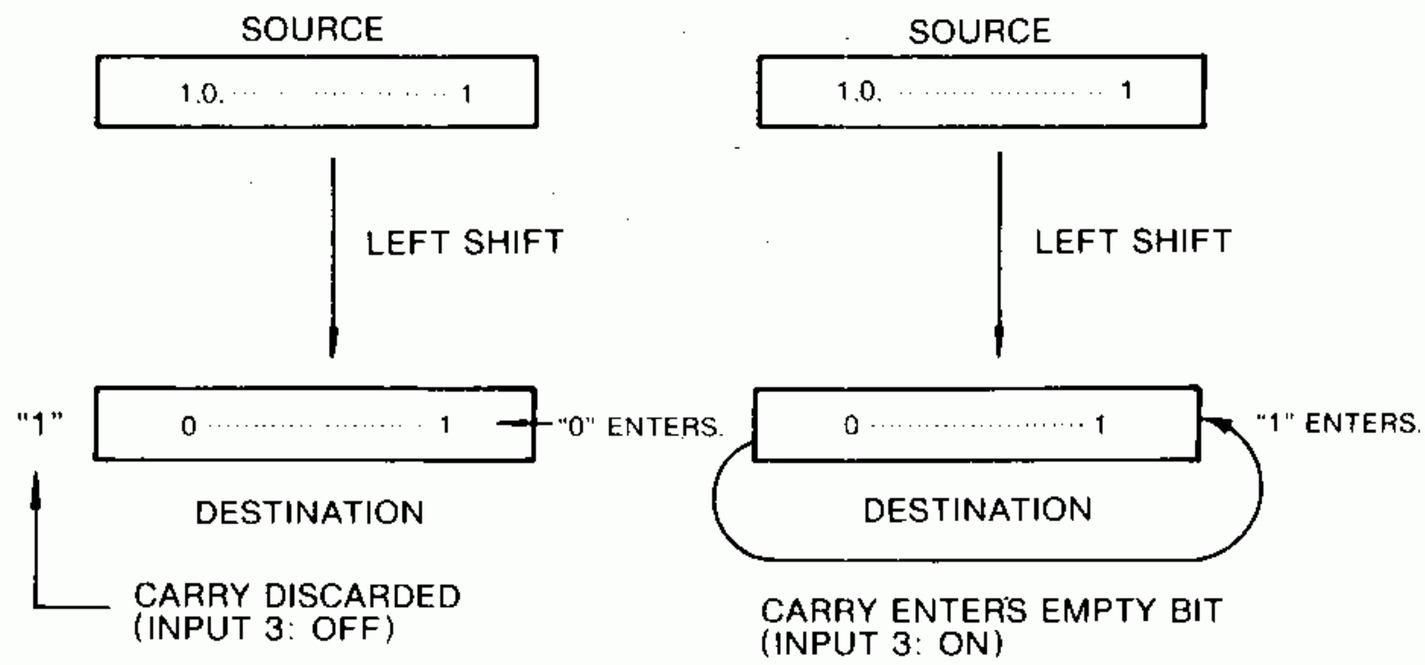
(b) Outputs

This function utilizes only the first two outputs. The output 3 has no significance and will be OFF (no power flow) under all conditions. The output 1 will supply power flow whenever the input 1 receives power flow. The output 2 supplies power flow when the carry is "1."

The direction of shift is determined by ON/OFF status of the input 2.



It is possible to select whether to fill the empty bit with 0 or with the carry, by ON/OFF status of input 3.



The source data remains unchanged unless the source and destination tables are the same. When the source and destination tables are the same, BROT realizes a 1-bit, N-stage (N is the total number of bits) shift register.

BROT function and status at I/O ON are shown in Tables 5.109 and 5.110, respectively.

Table 5.109 BROT Functions

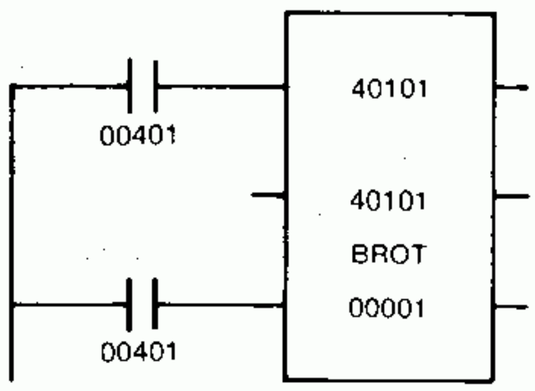
Input ×	Functions
Input 1	Executes BROT.
Input 2	• Shifts to the left (LSB→MSB) when it is ON. • Shifts to the right (MSB→LSB) when it is OFF.
Input 3	• When it is ON, a carried bit enters an empty bit. • When it is OFF, "0" enters an empty bit.

Table 5.110 BROT Output Status

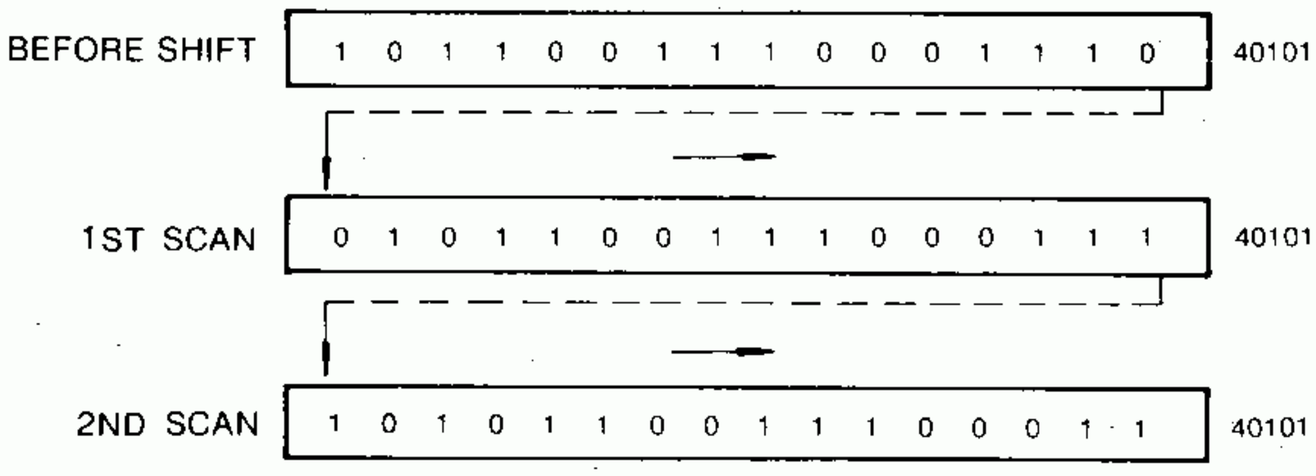
Output ×	Status
Output 1	Same as the input 1 ON/OFF status.
Output 2	It turns ON, when "1" is forced out as a result of shifting (carrier).
Output 3	Always OFF.

(4) Example

This is an example of a 1-bit, 16-stage ring counter. (The source and destination tables are the same.) This shifts to the right as the input 2 is OFF.



Ladder



5.12.9 Multi-Rotate (MROT)

(1) Function

MROT shifts all bits of the destination table, by the number of bits (1-15) stored in the source register to the left or right, and stores the result in the destination table all in a scanning cycle.

(2) Form

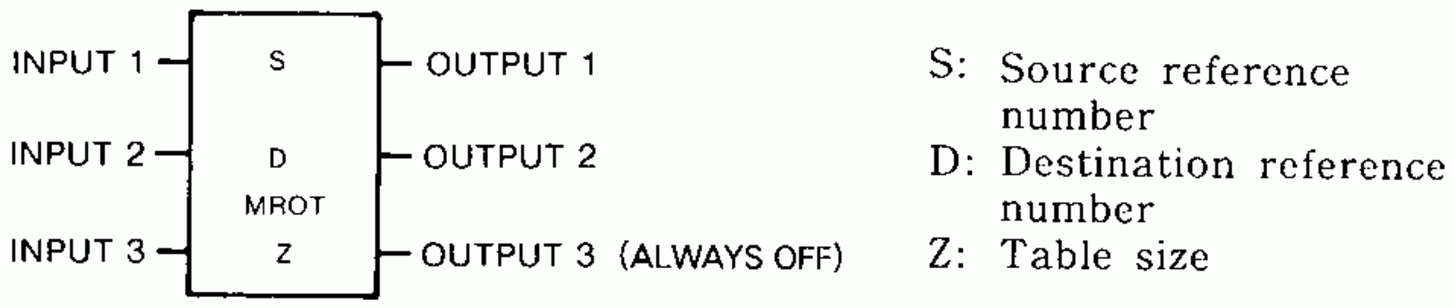


Fig. 5.88 MROT General Form

- Fig. 5.88 shows the form of multi-rotate.
- MROT is the symbol denoting multi-rotate.
- MROT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.111, specify the needed number for each element.

Table 5.111 MROT Elements

Element	Description	Specified Number
Top	Source reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

(3) Operation

(a) Inputs

When the input 1 receives power flow, MROT executes a shift. The input 2 determines the direction of shift as follows.

- When the inputs 1 and 2 are ON: Shift to left (to bit 1)
- When the input 1 is ON and 2 OFF: Shift to right (to bit N)
  The input 3 determines what should fill the empty bits as follows.
- When the inputs 1 and 3 are ON, the carry bits fill the empty bits.
- When input 1 is ON and 3 OFF, 0's fill the empty bit.

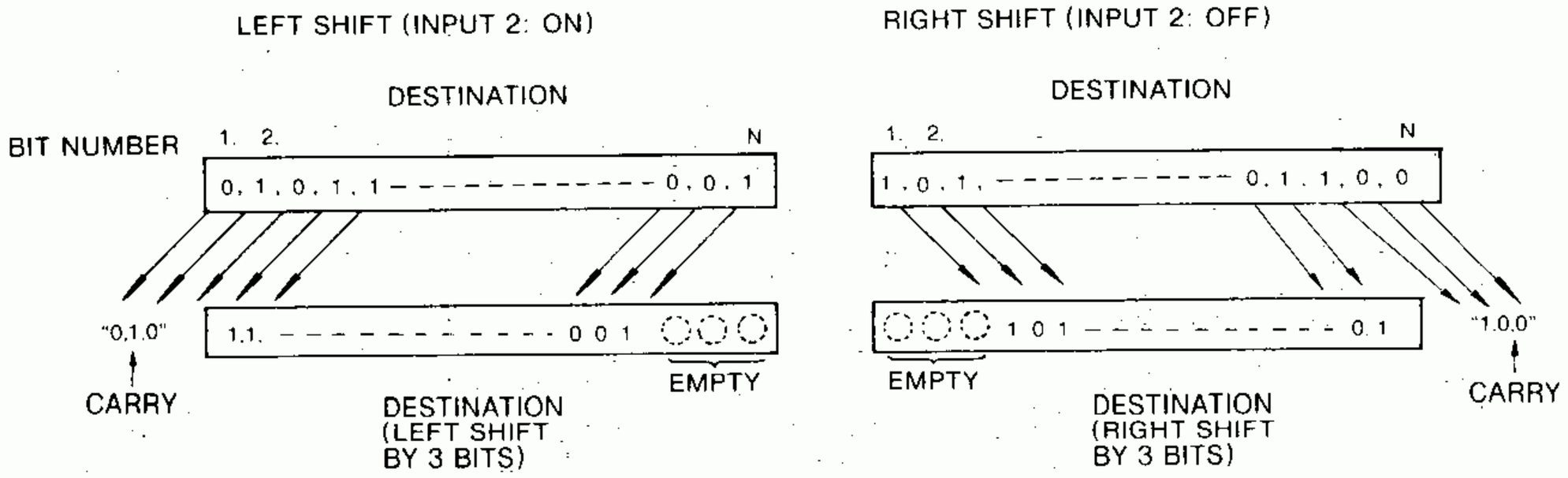
5.12.9 Multi-Rotate (MROT) (Cont'd)

(b) Outputs

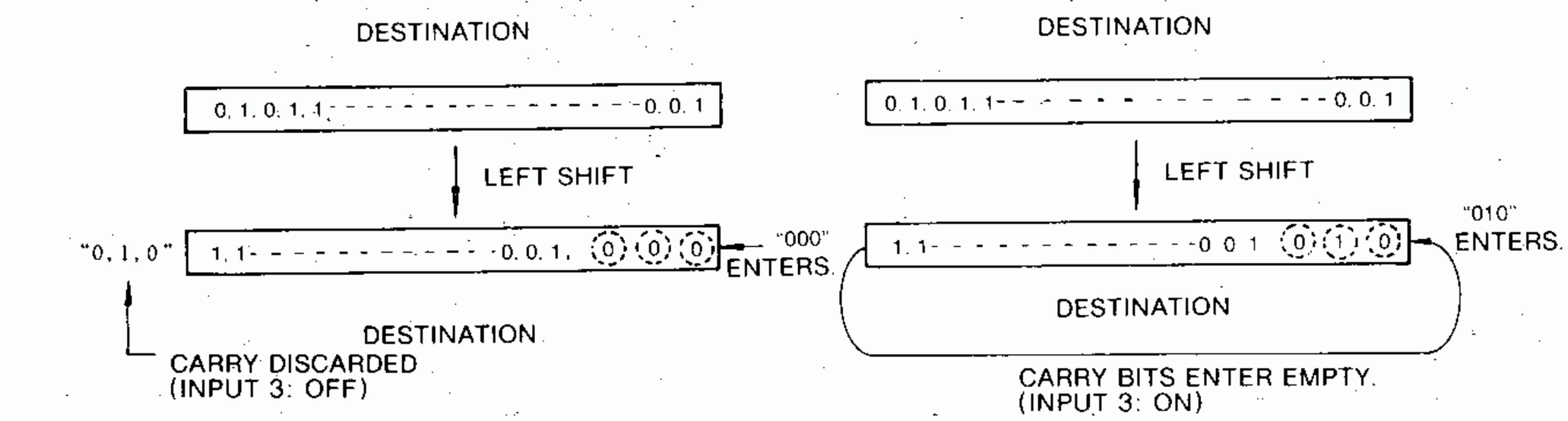
The output 1 is ON when the input 1 receives power flow and MROT executes a shift. The output 2 is ON when the input 1 receives power flow and MROT cannot execute a shift (the number of shift bits is 16 or more, or the source register is included in the destination table). The output 3 is always OFF.

Note When number of shift specified in source register is 0, the shift is not operated, but the output 1 is ON.

The input 2 determines the direction of shift.



It is possible to select whether to fill the empty bits with 0's or with the carry bits, by ON/OFF status of input 3.



Unlike BROT, the destination data are changed by the shift successively. MROT realizes an n-bit (n is the number of bits to shift), N-stage (N is the total number of bits) shift register.

MROT function and status at I/O ON are shown in Tables 5.112 and 5.113, respectively.

Table 5.112 MROT Functions

Input ×	Functions
Input 1	Executes MROT.
Input 2	• Shifts to the left (LSB→MSB) when it is ON. • Shifts to the right (MSB→LSB) when it is OFF.
Input 3	• When it is ON, a carried bit enters an empty bit. • When it is OFF, "0" enters an empty bit.

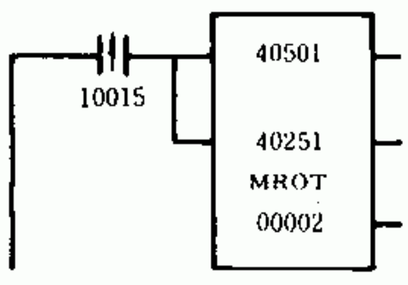
Table 5.113 MROT Output Status

Output ×	Status
Output 1	It turns ON when the input 1 is ON and MROT is executed.
Output 2	It turns ON when the input 1 is ON and MROT cannot be executed.
Output 3	Always OFF

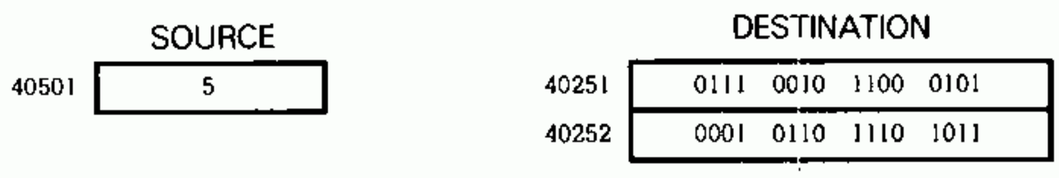
(4) Example

This is an example of 5-bit left shift. 0's fill the empty bits as the input 3 is OFF.

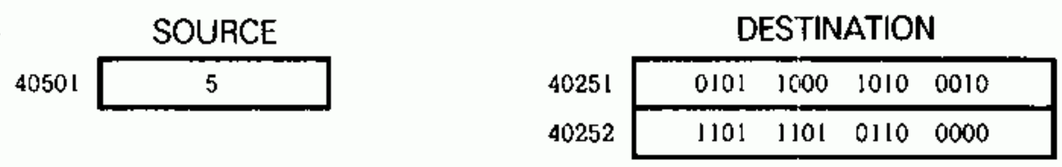
Example 1:



(a) Ladder



① Before Shift



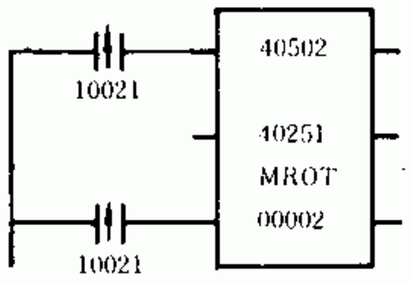
② After Shift

(b) Shift Operation

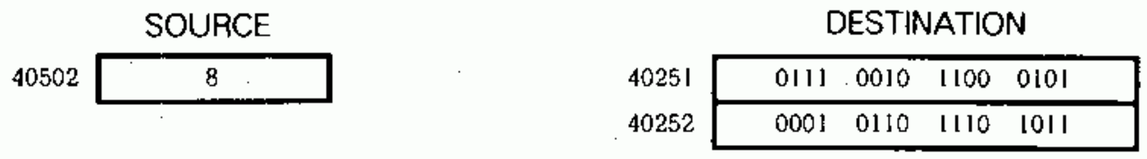
5.12.9 Multi-Rotate (MROT) (Cont'd)

This is an example of 8-bit right shift. The carry bits fill the empty bits as the input 3 is ON.

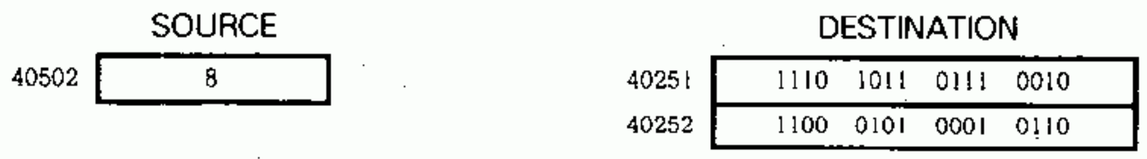
Example 2:



(a) Ladder



① Before Shift



② After Shift

(b) Shift Operation

5.12.10 Byte Rearrangement (TWST)

(1) Function

This function divides the registers in the destination table into the higher-place byte (8 bits) and the lower-place byte (8 bits). And bits in each byte are rearranged. All bits in the destination table are rearranged in one scan and stored in the destination table.

(2) Form

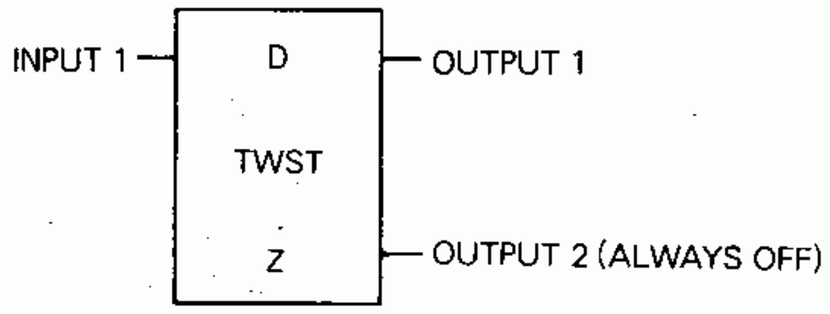


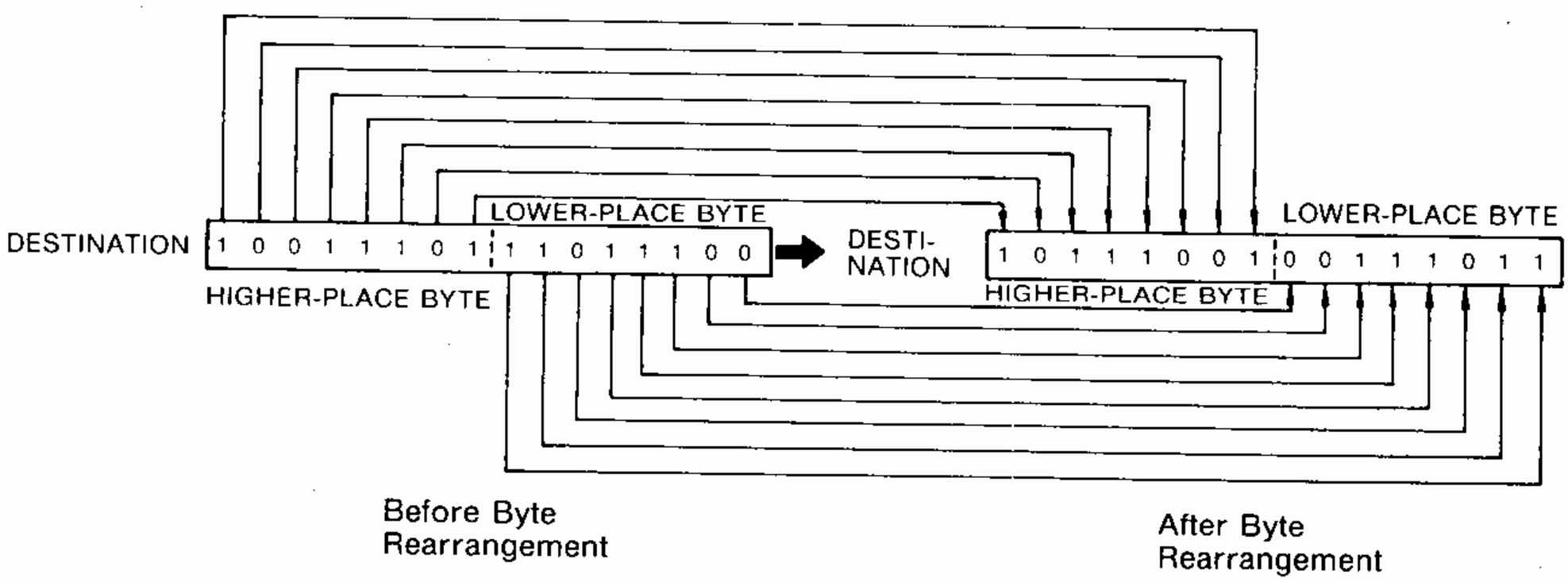
Fig. 5.89 TWST General Form

- Fig. 5.89 shows the form of byte rearrangement.
- TWST is the symbol denoting byte rearrangement.
- TWST requires two elements placed vertically (top and bottom). Referring to Table 5.114, specify the needed number for each element.

Table 5.114 TWST Elements

Element	Description	Specified Number
Top	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

(3) Operation



When the statuses of coils and input realys are read with the GL60S used as the MEMOBUS master, the information will come from a slave normally in such an order that a higher-place bit contains the status of a coil or input relay having a greater number. TWST can be used to rearrange the bits to the ordinary order.

SOURCE	
8 – 1	16 – 9
24 – 17	32 – 25
40 – 33	48 – 41

Before Byte Rearrangement

➡

SOURCE	
1 – 8	9 – 16
17 – 24	25 – 32
33 – 48	41 – 48

After Byte Rearrangement

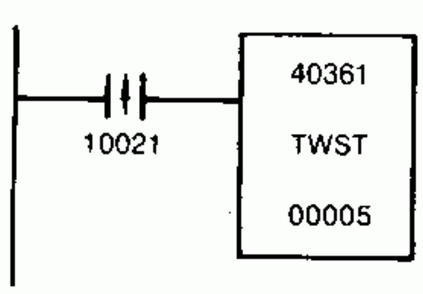
(a) Inputs

Only the input 1 is used for this function. When the input 1 receives power flow, the byte rearrangement operation can be performed.

(b) Outputs

TWST utilizes only the output 1 which will supply power flow whenever the input 1 receives power flow.

(4) Example



(a) Ladder

	DESTINATION	
40361	0110 1010	1100 0001
40362	1110 0101	1101 1110
40363	1110 0101	1100 1100
40364	1110 1110	1011 0101
40365	0101 0011	1011 1111

Before Byte Rearrangement

➡

	DESTINATION	
40361	0101 0110	1000 0011
40362	1010 0111	0111 1011
40363	1010 0000	0011 0011
40364	0111 0111	1010 1101
40365	1100 1010	1111 1101

After Byte Rearrangement

(b) Byte Rearrangement Operation

The higher-place bits can be replaced with the lower-place bits by using TWST and SWAP.

5.12.11 Bit Count (BCNT)

(1) Function

It counts the number of bits of "1" or "0" in the source table and stores the result in the destination.

(2) Form

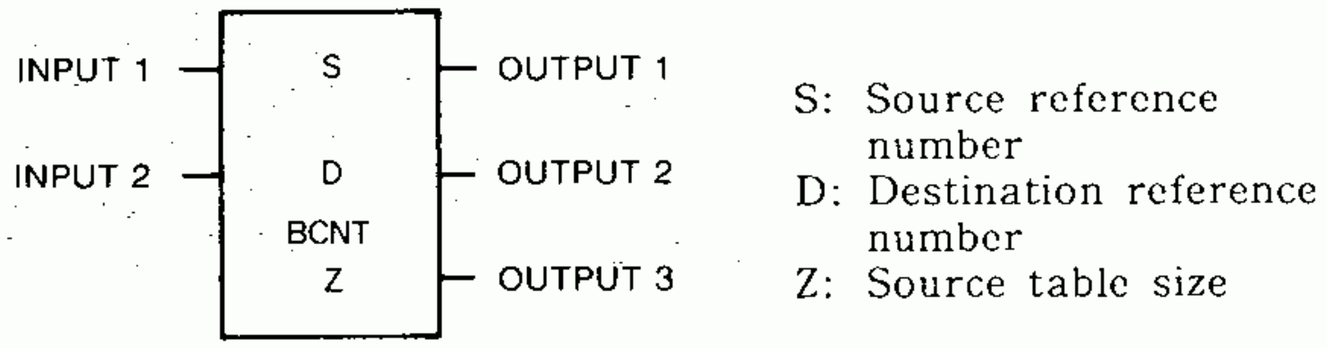


Fig. 5.90 BCNT General Form

- Fig. 5.90 shows the form of bit count.
- BCNT is the symbol denoting bit count.
- BCNT requires three elements placed vertically (top, middle, and bottom). Referring to Table 5.115, specify the needed number for each element.

Table 5.115 BCNT Elements

Element	Description	Specified Number
Top	Source reference number	• Input register (30001-30128) • Holding register (40001-42048) • Link register (R0001-R1024)
Middle	Destination reference number	• Holding register (40001-42048) • Link register (R0001-R1024)
Bottom	Table size	• Constant (1-100)

(3) Operation

When the input 1 turns ON, it counts the number of bits of "1" or "0" in the source table according to the ON/OFF status of the input 2 and stores the number in the destination table. It counts the numbers in all source tables in 1 scan.

BCNT function and status at I/O ON is shown in Tables 5.116 and 5.117, respectively.

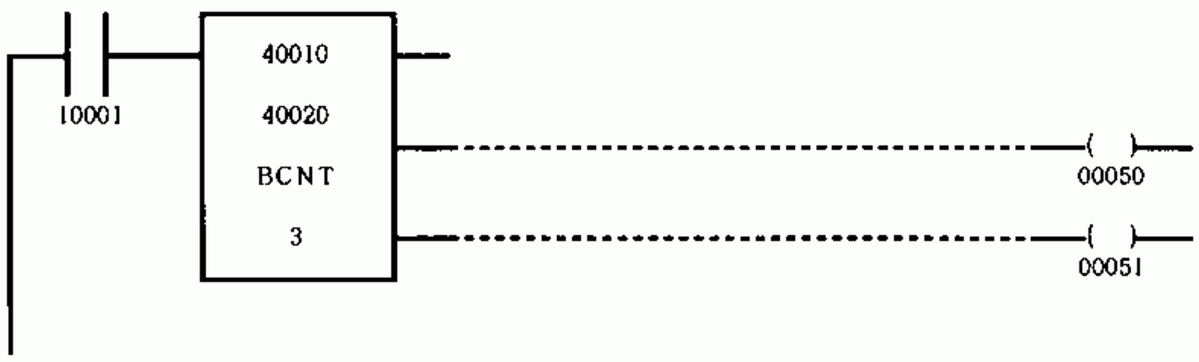
Table 5.116 BCNT Functions

Input ×	Functions
Input 1	Counts the number of bits of "1" or "0."
Input 2	Counts the number of bits of "1" at ON, and of "0" at OFF.

Table 5.117 BCNT Output Status

Output ×	Status
Output 1	Same as the input 1 ON/OFF status.
Output 2	It turns ON when the number of bits of "1" or "0" is an odd number.
Output 3	It turns ON when the number of bits of "1" or "0" is an even number.

(4) Example



(a) Ladder

SOURCE TABLE

40010	1100	1010	0001	1001
40011	0101	1000	0000	0011
40012	1111	0000	1111	0000

DESTINATION

40020	1000

① Before BCNT

SOURCE TABLE

40010	1100	1010	0001	1001
40011	0101	1000	0000	0011
40012	1111	0000	1111	0000

DESTINATION

40020	28

② After BCNT

(b) Bit Count Operation

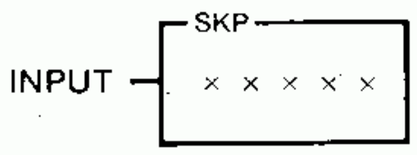
BCNT shown in (a) counts the number of "0" bits of the source table, because the input 2 turns OFF when input relay 10001 turns ON. Since the operation result is 28, which is an even number, the output 3, that is, output coil 00051 turns ON. This function can be used for parity check of data.

5.13 SKIP

This function allows logic in groups of networks to be skipped and thus not sloved. Networks that are skipped will have their coils (if any) unchanged and register content unaltered; skipped networks are thus basically "frozen."

The Skip Function is useful to reduce the logic scan time. A skip of zero networks saves all of the time required to solve the logic remaining. A skip of non-zero networks reduces the time to solve the skipped logic to that of relay functions. The reduction in scan time cannot exceed that of the logic remaining. Since all logic must be examined to count and determine quantity of networks (whose size does vary greatly), non-zero skips will save less time than a zero skip. In either case, the controller's over all scan time cannot be reduced beyond that required to service I/O devices. In addition, the Skip Function can be used to select various groups of logic to be performed from a larger selection of logic.

(1) Form



Only one function block is used with the skip function, and input is available for control

- Reference number

  The function block specifies the quantity of networks to be skipped. The value can be fixed quantity up to 9999 or a register reference (3× × × × or 4× × × ×).

(2) Function and Operation

Assume the Skip instruction is stored in the network N and the number of networks to be skipped is J.

In a scanning cycle when the input is ON, processing of the J successive networks including network N (N, N+1, N+2, ..., N+J-1) is stopped.

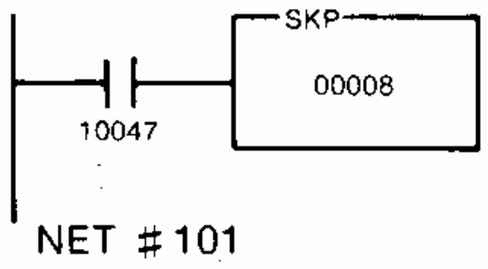
If J = 0 or J is greater than the number of networks following network N, processing of all networks following network N will be skipped.

In network N, all logics that follow the element of Skip according to the order of network solving (see Par. 4.5.1) will be skipped.

Note 1. In case segment allocation is made in 2-level scan, no Skip is allowed exceeding the segments.

2. Use of this function in action circuit or subroutine circuit is not allowed.

(3) Circuit Example

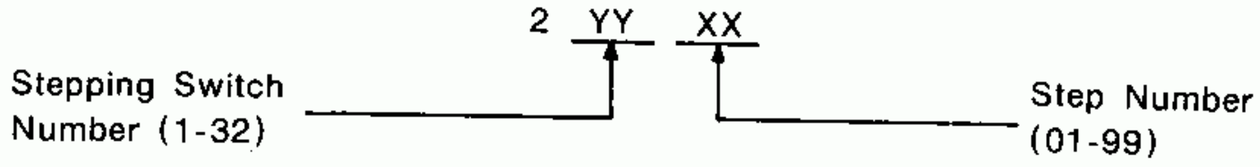


When the input relay 10047 is ON, networks 101-108 will be skipped. If the network 105 is the last one, networks 101-105 will be skipped.

5.14 STEPPING SWITCH

5.14.1 Stepping Swicth Functions and Operation

The GL40S Controller is provided with thirty-two independent 99-step stepping switch. These references start with the digit 2 in the form 2YYXX. The significance of the remaining three digits of the reference is as follows:



The stepping switches one to thirty-two are controlled by numerical values placed in registers 42001-42032, resperctively. The numerical values can be placed in these specific registers (called stepping switch control registers) by any of the non-relay functions such as counter, timers, or any arithmetic operations. A value of zero or above 32 will result in all references to that stepping switch being de-energized and all other references to that stepping switch being de-energized.

Fig. 5.91 shows an equivalent circuitry to stepping switch 1. As shown in the figure, stepping switch 1 operates according to the holding register 42001 contents (n), as described below:

- When n is in the range of 1 to 99:

  Assume that any one of coils 20101 to 20199 is turned ON corresponding to the value of n. NO contact is conductive and NC contact is nonconductive. For example, when n is 1, only coil 20101 is turned ON; NO contact is conductive and NC contact is nonconductive.

- When n is outside the range of 1 to 99:

  All coils 20101 to 20119 are turned OFF.

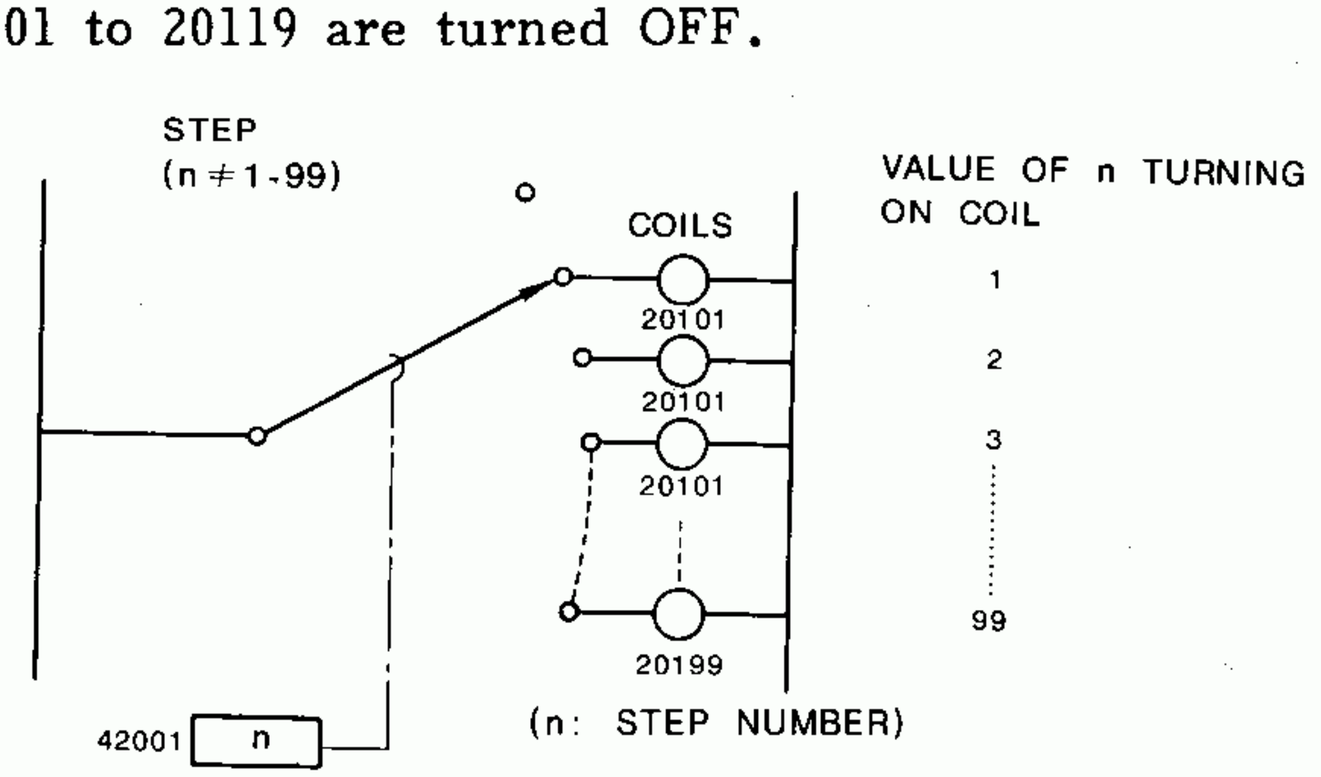


Fig. 5.91 Equivalent Circuitry to Stepping Switch 1

5.14.1 Stepping Swicth Functions and Operation (Cont'd)

Table 5.118 lists operation of stepping switches 1 to 32.

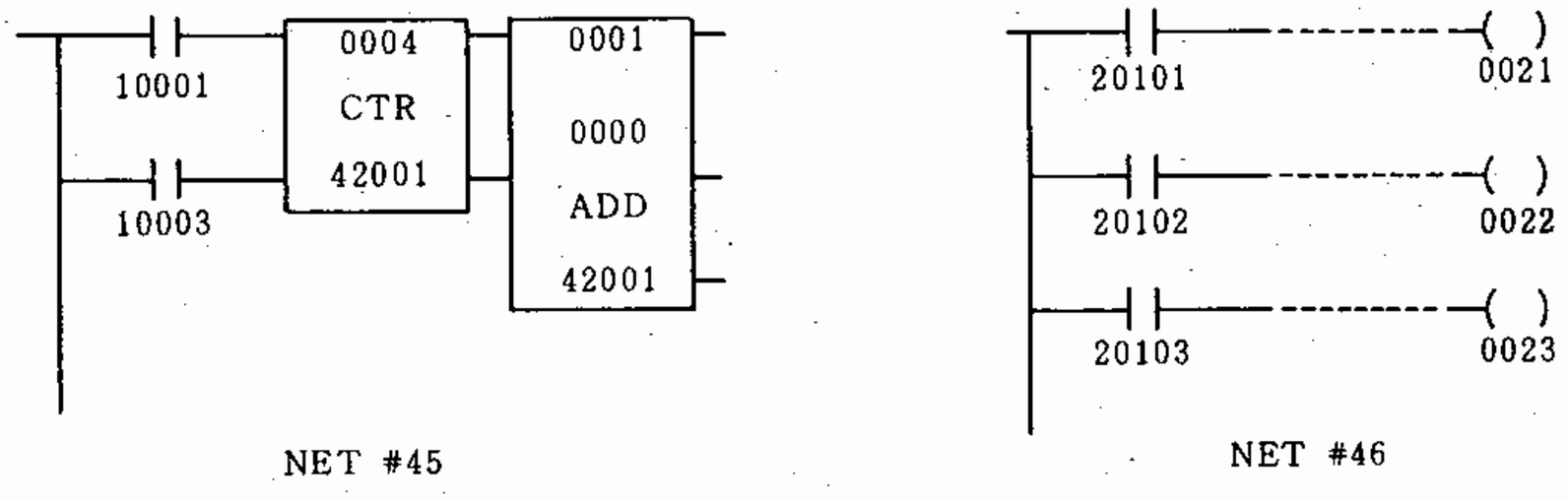
Table 5.118 Stepping Switch Operation

Stepping Switch No.	Stepping Switch Control Register No.	Stepping Switch Control Register Contents (Numeric Value)				
		001	002	003	·····	99
1	42001	20101 "ON"	20102 "ON"	20103 "ON"	·····	20199 "ON"
2	42002	20201 "ON"	20202 "ON"	20203 "ON"	·····	20299 "ON"
3	42003	20301 "ON"	20302 "ON"	20303 "ON"	·····	20399 "ON"
4	42004	20401 "ON"	20402 "ON"	20403 "ON"	·····	20499 "ON"
5	42005	20501 "ON"	20502 "ON"	20503 "ON"	·····	20599 "ON"
⋮						
31	42031	23101 "ON"	23102 "ON"	23103 "ON"	·····	23199 "ON"
32	42032	23201 "ON"	23202 "ON"	23203 "ON"	·····	23299 "ON"

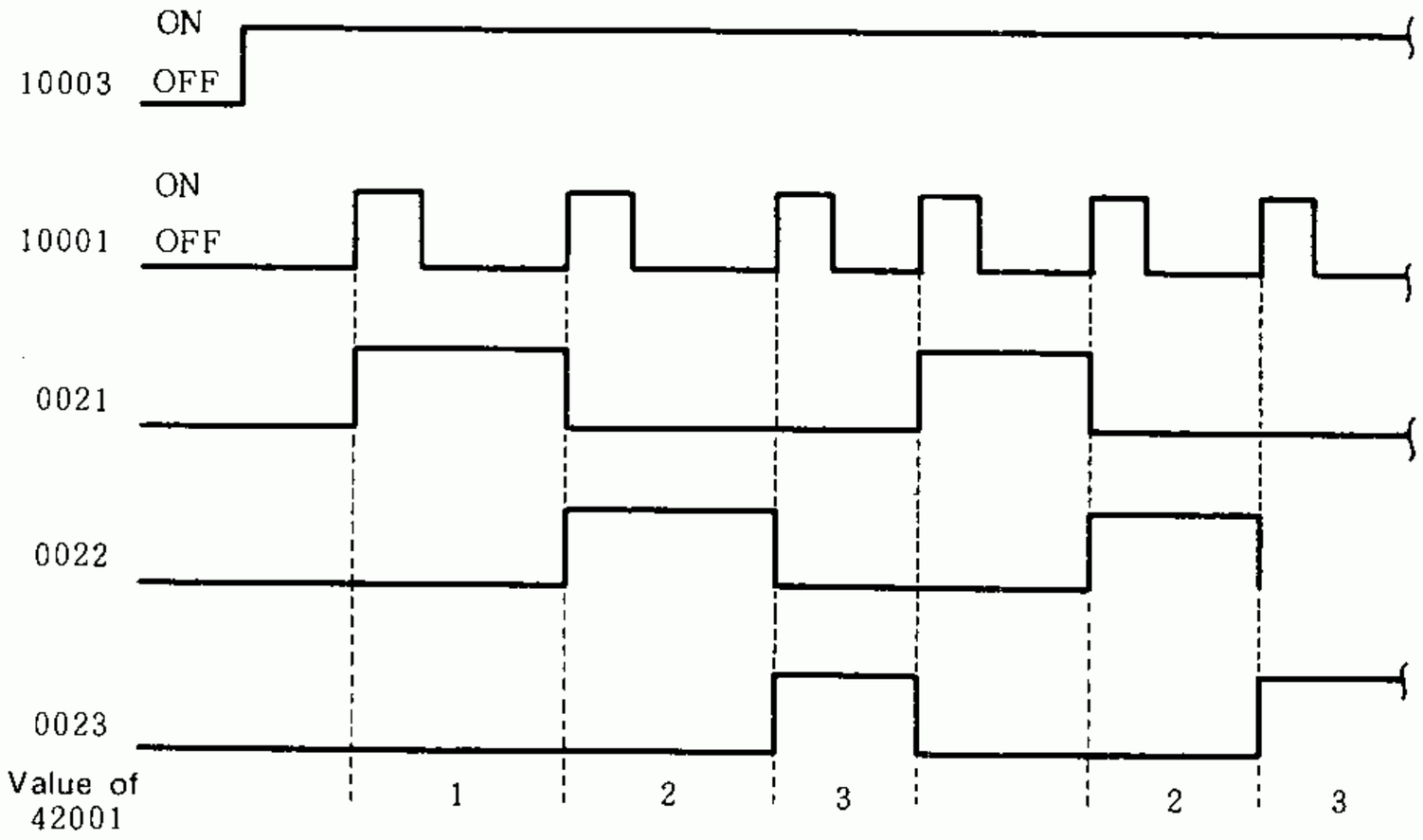
Note
1. If all switches are not being used, the registers 42001-42032 can be used exactly as any other holding register.
2. If stepping switches are used in a network, their NO contacts and NC contacts are used. Transitional contacts cannot be used.

5.14.2 Examples of Stepping Switch Use

(1) As shown in Fig. 5.92, stepping switch 1 is caused to proceed step by step by using a counter. In (a) Ladder Diagram, coils 21 to 23 operate as shown in (b).



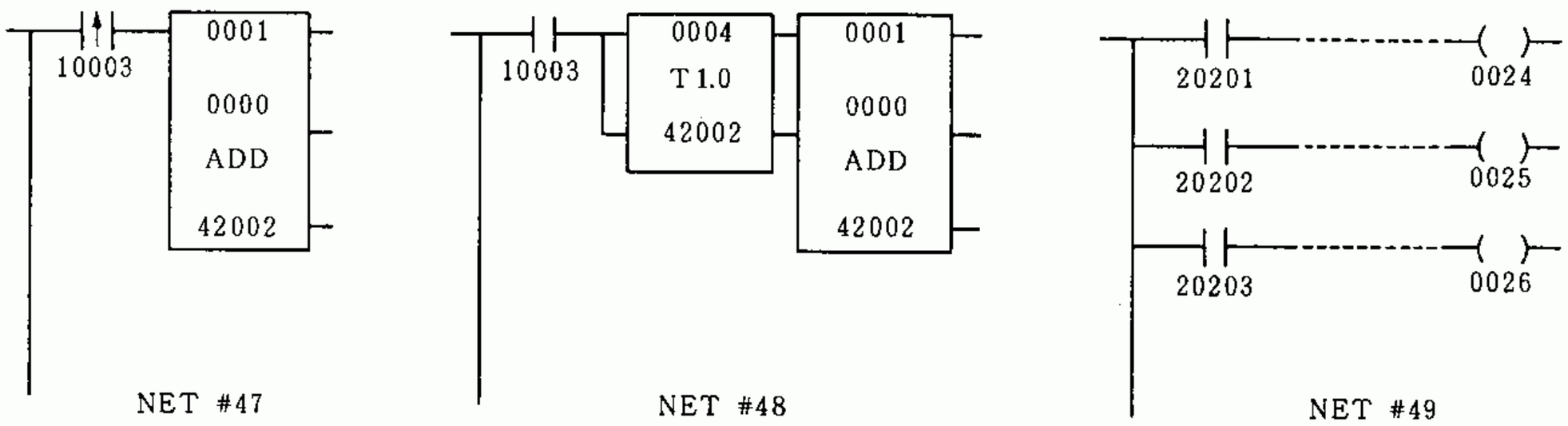
(a) Ladder Diagram



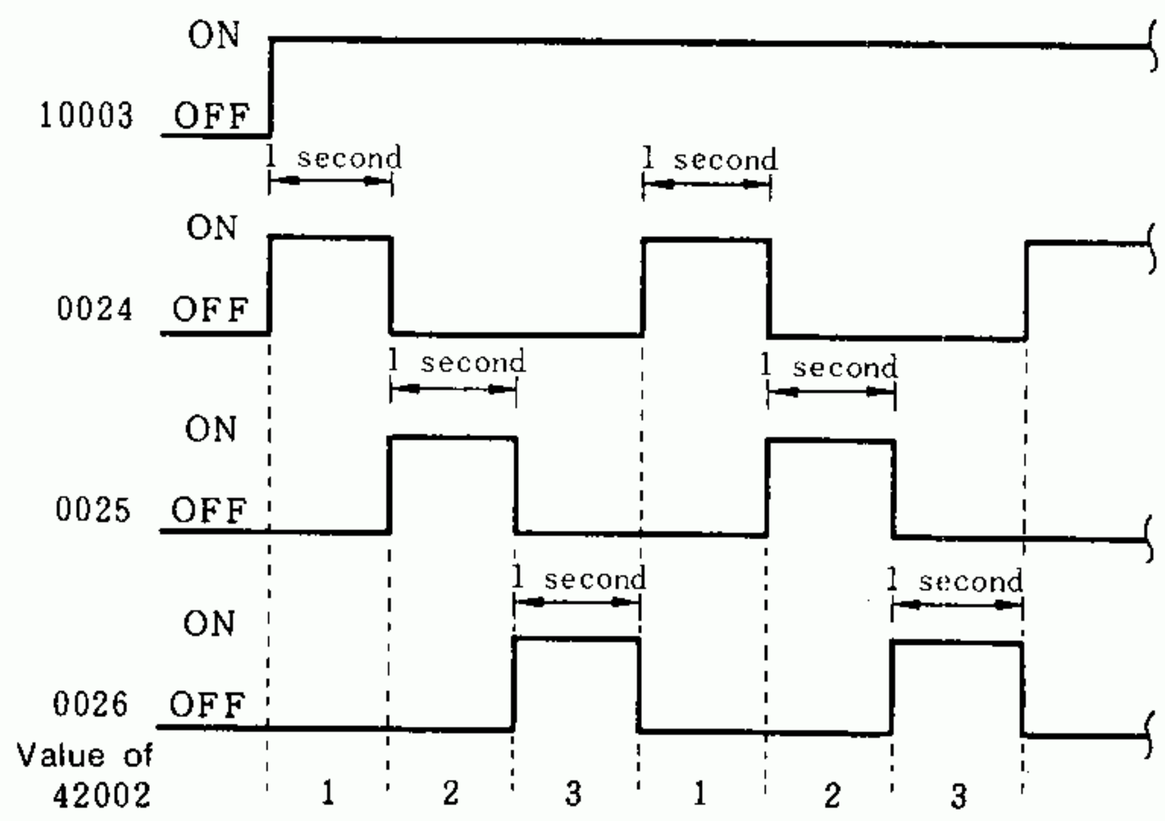
(b) Operation

Fig. 5.92 Example 1 of Stepping Switch Use

(2) As shown in Fig. 5.93, stepping switch 2 is caused to proceed step by step by using a timer. In (a) Ladder Diagram, coils 24 to 26 operate as shown in (b).



(a) Ladder Diagram



(b) Operation

Note Consider "Timer error" in par. 5.3.4 (7).

Fig. 5.93 Example 2 of Stepping Switch Use

5.15 SUBROUTINE

(1) Function

When the same circuit is to be formed in a ladder circuit many times (if it is stored as a subroutine circuit), it can be freely called up from the ladder circuit, and only one circuit has to be formed.

(2) Form

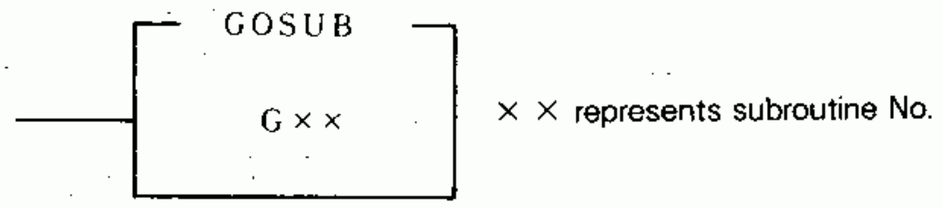
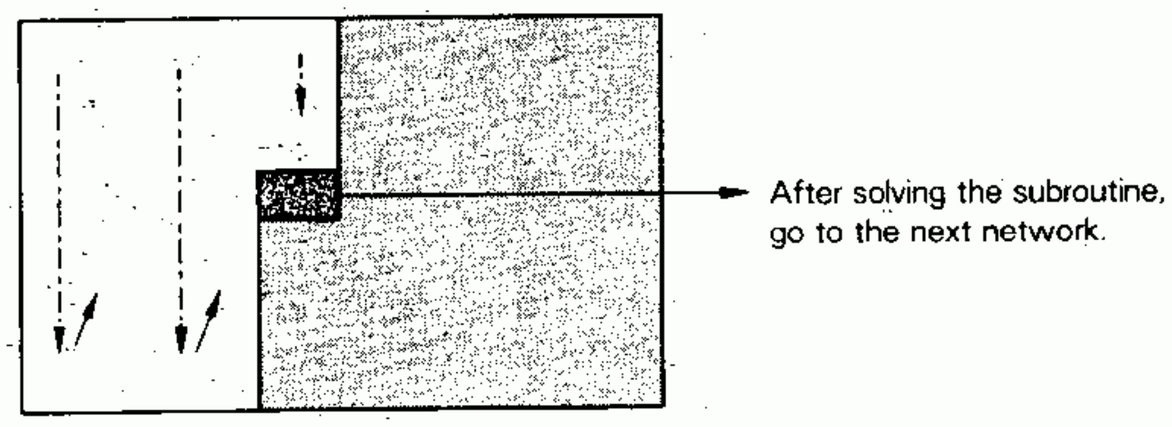


Fig. 5.94 GOSUB General Form

- Fig. 5.94 shows the form of subroutine.
- GOSUB is the symbol denoting subroutine.
- GOSUB requires subroutine No. as a component element. Specify a constant in the range form 0 to 99 according to the service condition of the subroutine circuit. No output is made.

(3) Operation

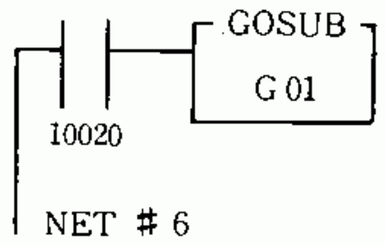


In the above diagram, suppose that GOSUB is at the position of [redacted]. At this time, when solving is performed in sequence, the input of GOSUB is ON and GOSUB is solved, solving jumps to the specified subroutine circuit.

When solving of the subroutine circuit is finished, Solving will start from the head of the network next to the network in which GOSUB had been included. Consequently, since the part shown with [redacted] in the above diagram will not be solved, any attempt to input coil or relay from P150 Programming Panel will fail. For details see "P150 Programming Panel Manual".

Furthermore, the same subroutine circuit can be called up any number of times by GOSUB, but in a subroutine circuit, it is not possible to further call up a subroutine by GOSUB (the so-called nesting). A subroutine circuit is stored in the unit of network, and it is formed within the scope of memory allocation of subroutine.

(4) Example



When input relay 10020 is ON, the subroutine circuit with subroutine No. 01 will be solved. Upon completing solving, this function restarts solving of the ladder circuit from NET #7.

The subroutine circuit with subroutine No. 01 makes programs from the network #1 to the specified network #n with the same procedure for ordinary networks.

Note For the state of coil used in the subroutine circuit and the content of register, in the case of the scan where the subroutine is not called up by GOSUB, the state in the scan called up last is held. Consequently, when the coil which is in the ON state is required to be turned OFF, the subroutine must be called up once again, after making an arrangement to allow the coil to turn OFF.

5.16 COMMUNICATION COMMAND

5.16.1 Features and System Configuration

5.16.1.1 Features

With COMM command, communication can be activated from the GL40S side. Communication can be performed via expanding communication module (COMM) RS-232C communication port as MEMOBUS master unit.

Communication with other devices using different protocol from MEMOBUS is also possible by application program.

5.16.1.2 System Configuration

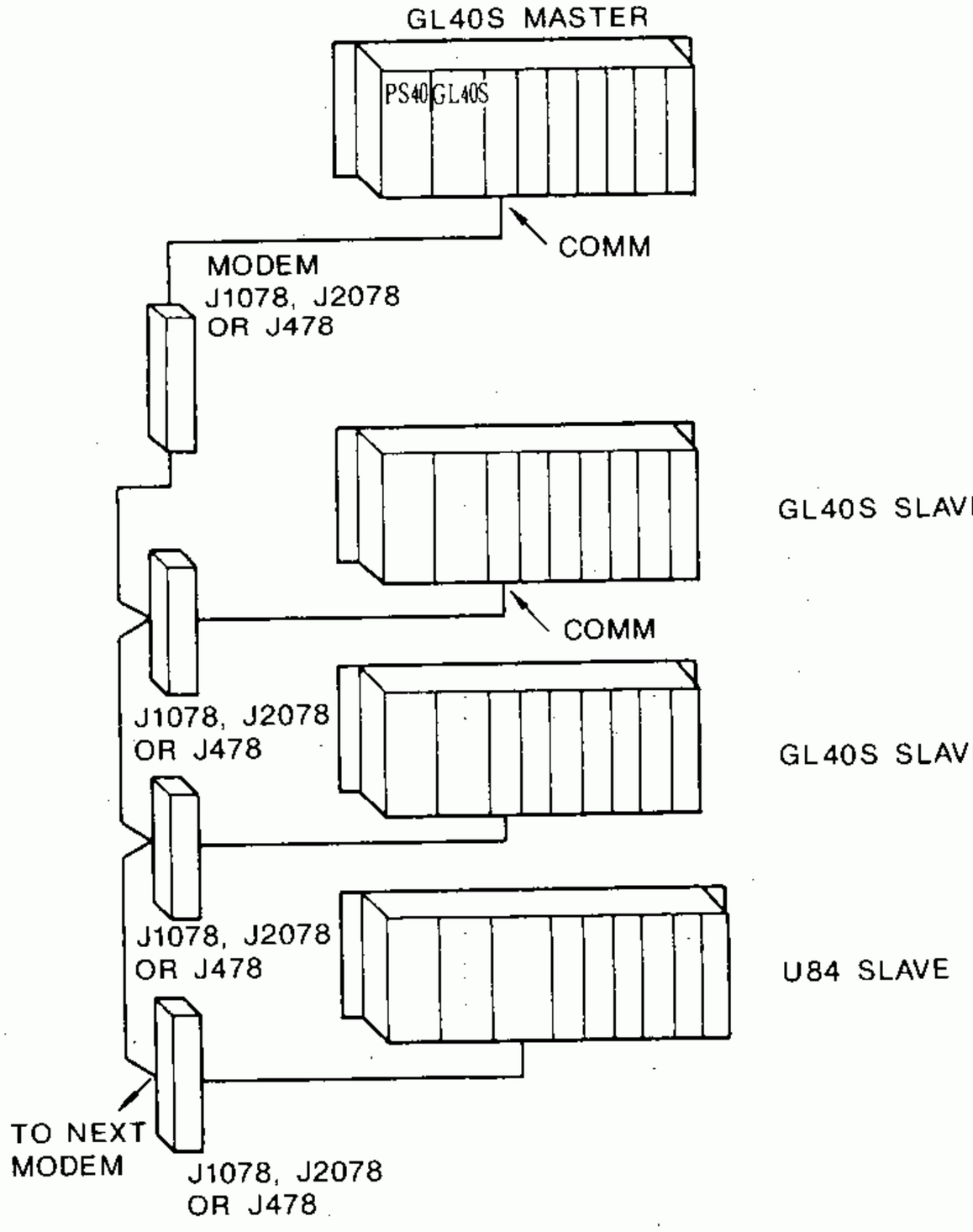


Fig. 5.95 GL40S MEMOBUS Master Unit

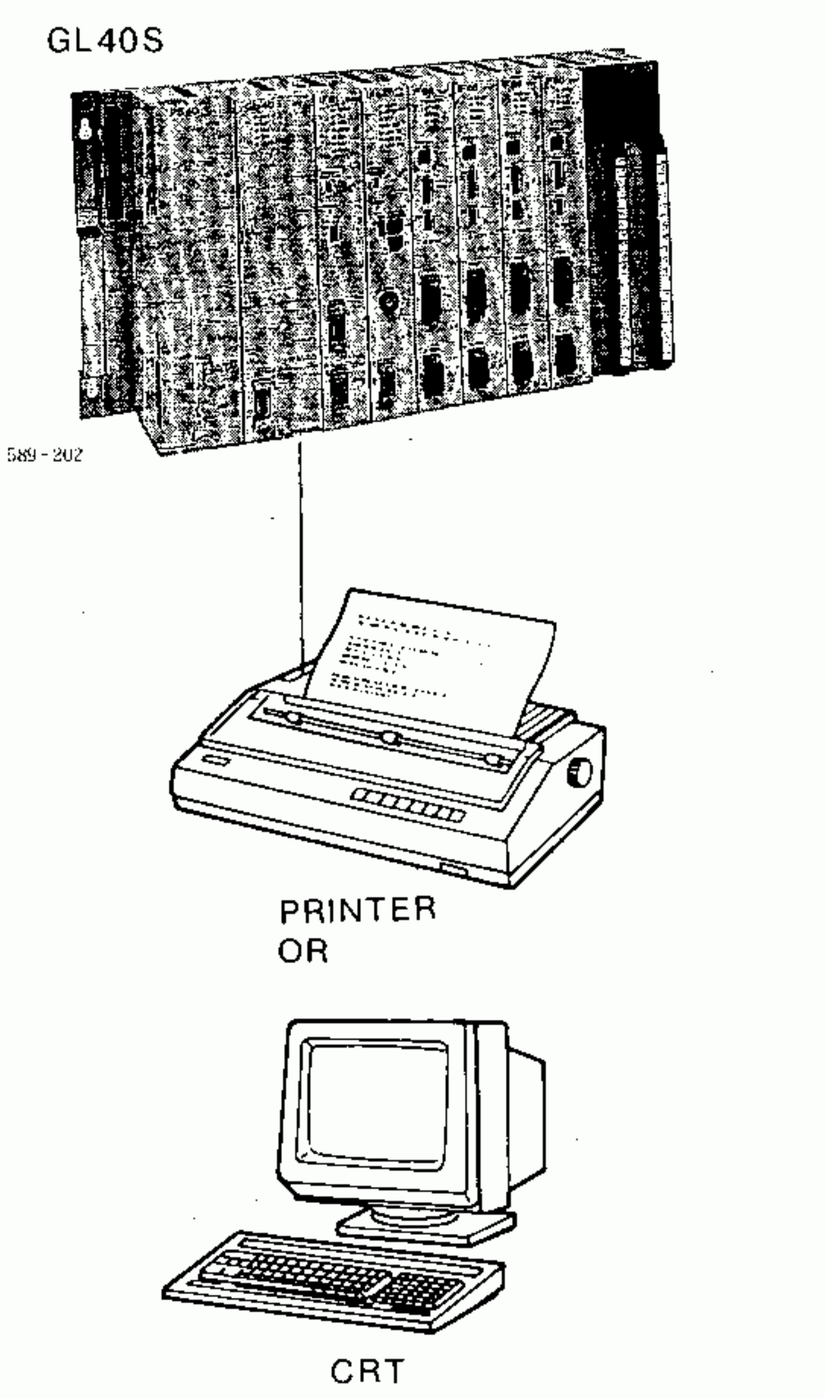


Fig. 5.96 Connection with Other Devices

5.16.2 Specifications

5.16.2.1 Transmission Specifications

COMM board specifications are shown below as transmission specifications.

Table 5.119 Transmission Specifications

Item	Specification
Standards	Conform to EIA RS-232C.
Synchronization	Half-duplex asynchronous
Transmission Speed	19200/9600/4800/2400/1200/600/300/150 baud
Quantity of Characters	RTU (Remote Terminal Unit): 8 bits, ASCII: 7 bits
Stop Bit	1 or 2 bits
Parity Check	Even, odd or no parity
Connector	Mini D-sub 9-pin connector

5.16.2.2 Required Time for Signal Transmission

Required time for a signal transmission by using COMM command can be calculated from the following seven related times. If more than one slave is connected to the same port on the master, calculate the required time for a signal transmission for each slave and add the results.

(1) Query Message Transmission Processing Time by Master

This time interval comprises the time during which the master constructs a query message and then prepares it at a communication port after COMM command is activated. The time interval depends on the scan time of each master.

Note that only the first COMM command to be read is in execution to the same port though more than one COMM command is possible to be activated simultaneously. The other commands are in waiting status. After execution, the next command is executed. Therefore, the value is given when only one command is activated.

(2) Delay Time of Modem (Master Side)

This time interval comprises the time from the modem receiving RTS (request to send) from the master to the time when the modem sends CTS (clear to send) to the master. With modems J478, J1078 and J2078 (YASKAWA's standard modems) the delay time is up to 5 ms.

In a system not using any modem, this time is not included in the calculation.

(3) Query Message Transmission Time

To calculate the time, use the equation below. Number of bits per character is the sum of the number of data bits (7 or 8 bits), start bits (1 bit), stop bits (1 or 2 bits), and parity bits (0 or 1 bit).

$$\text{Transmission Time} = \frac{\text{Number of characters of query message} \times \text{Number of bits per character} \times 1000}{\text{Baud rate}} \text{ (ms)}$$

(4) Slave Processing Time

This time interval comprises the time from a slave receiving and processing a query message sent from the master to the time when the slave prepares a response message to the master at the communication port.

The slave processing time for MEMOBUS is shown as follows.

Table 5.120 Maximum Quantity to be Processed by One Scan

Function Code (Decimal)	Function	U84 584	R84H-M GL20	GL60S	GL40S
1	Coil state read-out	64	32	2000**	2000**
2	Input relay state read-out	64	32	2000**	512**
3	Holding register content read-out	64	16	125	125
4	Input register content read-out	32	16	125	64
5***	Single-coil state change	1*	1*	1*	1*
6***	Write-in to single holding register	1	1	1	1
7	Paticular coil state read-out	8	8	8	8
8	Loopback test	—	—	—	—
15***	Multi-coil state change	64*	—	800	800
16***	Write-in to holding register	32	—	100	100
18	Specified link relay state read-out	—	—	1024	1024
19	Constant register content read-out	—	—	125	—
20	Step passed time content read-out	—	—	125	—
21	Link register content read-out	—	—	125	125
22	Extension register content read-out	—	—	125	—
23	Step state read-out	—	—	512	—
25***	Specified link relay state change	—	—	1*	1*
26***	Write-in to specified constant register	—	—	1	—
27***	Write-in to specified link register	—	—	1	1
28***	Write-in to specified extension register	—	—	1	—
29***	Multi-link relay state change	—	—	1024*	1024*
30***	Write-in to multi-constant register	—	—	123	—
31***	Write-in to multi-link register	—	—	123	123
32***	Write-in to multi-extension register	—	—	123	—

* Two scans are required.
** Scan will be 10 ms longer for the maximum value.
*** "0" can be specified as slave address.

The following processing times are not necessary when only transmission is performed.

(5) Delay Time of Modem (Slave Side)

This time interval consists of the time from the modem receiving RTS from a slave to the time when the modem returns CTS to the slave. In the case of J478, J1078 and J2078 modems the delay time is up to 5 ms.

In a system not using any modems, this time is not included in the calculation.

(6) Response Message Transmission Time

The following equation, which is the same as that of the calculation of query message transmission time, should be used.

$$\text{Transmission Time} = \frac{\text{Number of characters of query message} \times \text{Number of bits per character} \times 1000}{\text{Baud rate}} \text{ (ms)}$$

(7) Response Message Processing Time by Master

1 to 2 scans

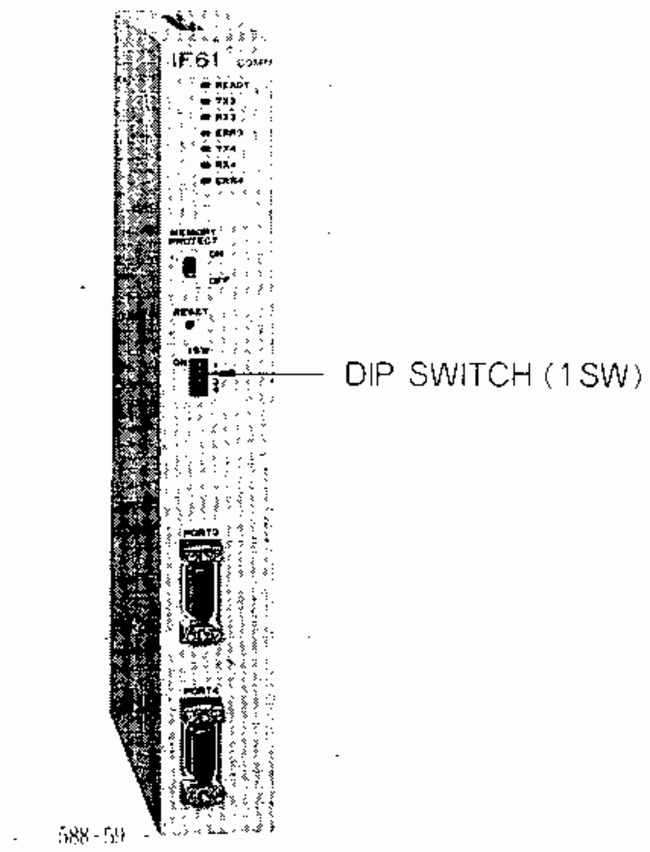
This time interval is required when master GL40S receives response message from the slave and process is performed to complete the COMM commands.

5.16.3 Communication Modes

5.16.3.1 Setting Communication Modes

There are two units between the master and slave units: MEMOBUS and transparent modes (see par. 5.16.3.3). Communication modes are set by dip switch (1SW) of COMM. Table 5.121 shows a list of settings.

These settings are valid only when COMM commands are used for communication.



COMM (IF61 or IF41A)

Fig. 5.97 Dip Switch(1SW)

Table 5.121 COMM

1SW	Port No.	Setting	Communication Mode
2	3	ON	Transparent mode
		OFF	MEMOBUS mode
3	4	ON	Transparent mode
		OFF	MEMOBUS mode

1SW−1 and −4 should be set to OFF.

Note COMM commands for GL40S are available for Port 3 and 4. COMM commands can not be used for Port 1.

5.16.3.2 Setting Communication Parameters

Set the communication parameters with RAP (register access panel) or programming (P150). The communication parameters set when GL40S system is activated are used. Change can be possible at any time.

Table 5.122 Communication Parameters

Baud Rate	150, 300, 600, 1200, 4800, 9600, or 19200
Parity	Parity check disabled, even parity or odd parity
Stop Bit	1 or 2
Communication Mode	RTU or ASCII

5.16.3.3 Transparent Mode

In this mode, transmission data prepared in the holding register are sent without being processed and received data from the port are stored in the holding register. Accordingly, no communication protocol is specified so that flexible communication is available by application programs. However, BCC (binary check code) of CRC (cycle redundancy check) or LRC (longitudinal redundancy check) is not made automatically by COMM: it must be made and added by application programs.

There are some differences between RTU mode and ASCII mode as follows.

(1) RTU Mode (8-bit Mode)

At transmission, the holding register data specified by command are sent from the port. Upon reception, the received data are input in the specified holding register.

Completion of query message at transmission/receiving is performed by timer (24-bit time).

(2) ASCII Mode (7-bit Mode)

At transmission, the holding register data specified by command are sent from the port without changing codes. In case of ASCII mode, the number of data bits is 7. MSB is not sent out.

Upon reception, there are some definitions of query message formats from the remote station. Completion of query message at receiving is performed by an identifier for completion (CR: carriage return, LF: line feed) or timer (24-bit time). All transmission data including the identifier for completion are input in the holding registers specified by the command.

5.16.3.4 MEMOBUS Mode

This mode is used when GL40S is used as MEMOBUS master unit.

At transmission, the data applicable to MEMOBUS protocol are prepared as transmission data in the holding register specified by the command. BCC is added by COMM. Therefore, the received data other than BCC are input in the specified holding registers.

In this case, set different value between master device address and slave device address. If same value are set, communication error will occur.

(1) RTU Mode

At transmission, the holding register data specified by the command are sent from the port. Upon reception, the received data are input in the specified holding registers.

Completion of the query message at transmission/receiving is performed by timer (24-bit time). The number of receiving data items does not include BCC (in this case, BCC of CRC).

(2) ASCII Mode

At transmission, the holding register data specified by command are changed to 8-bit ASCII code. The number of transmission data is 7.

A start identifier ":" and a completion identifier "CR, LF" of the query message are added automatically by COMM at transmission.

Upon reception, the start and completion identifiers are removed automatically and remaining data are stored in the holding register specified by the command as received data. At this time, the data are reversed from ASCII code.

The number of received data items does not include BCC (in this case, BCC of LRC).

Fig. 5.98 shows the relation of register data and actual transmission/receiving data for the ASCII mode.

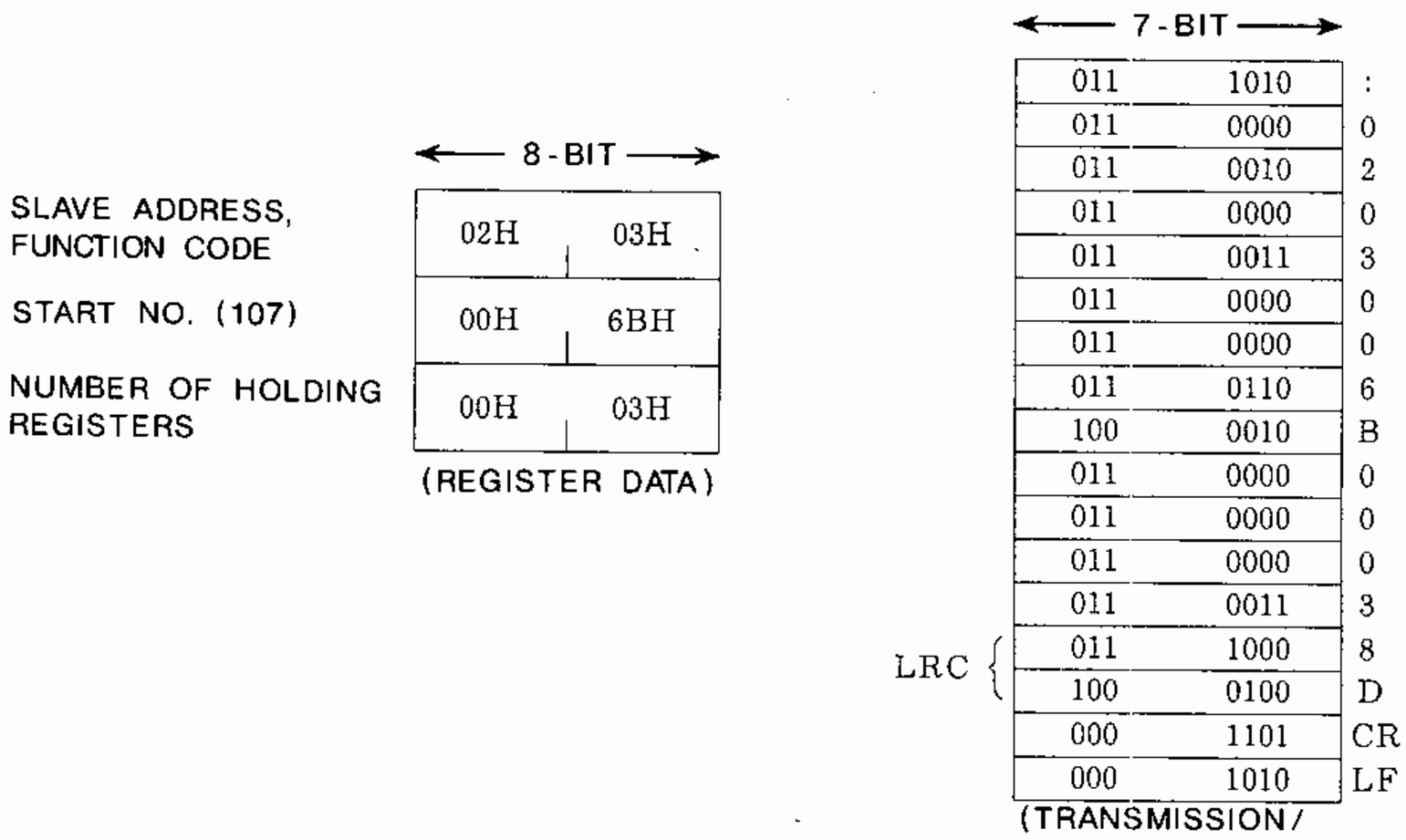


Fig. 5.98 Register Data and Transmission/Receiving Data

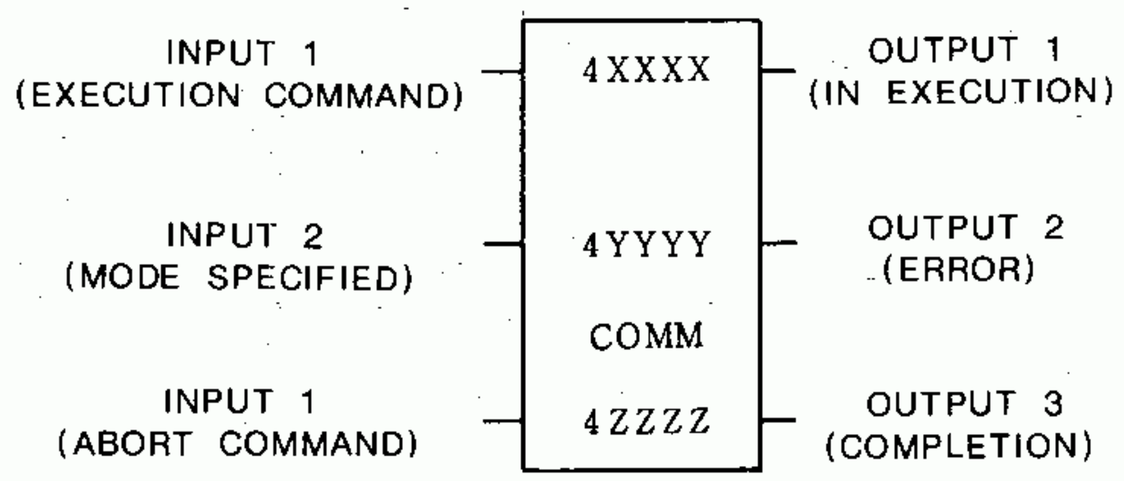
5.16.3.5 COMM Commands

Mode	COMM Commands			
	Transparent		MEMOBUS	
	RTU	ASCII	RTU	ASCII
Identifer for start	—	—	—	: (colon)
Identifer for completion	24-bit time	CR, LF or 24-bit time	24-bit time	CR, LF
BCC	—	—	CRC	LRC

5.16.4 Command

All transmission query messages are prepared and received query messages are stored in the holding registers.

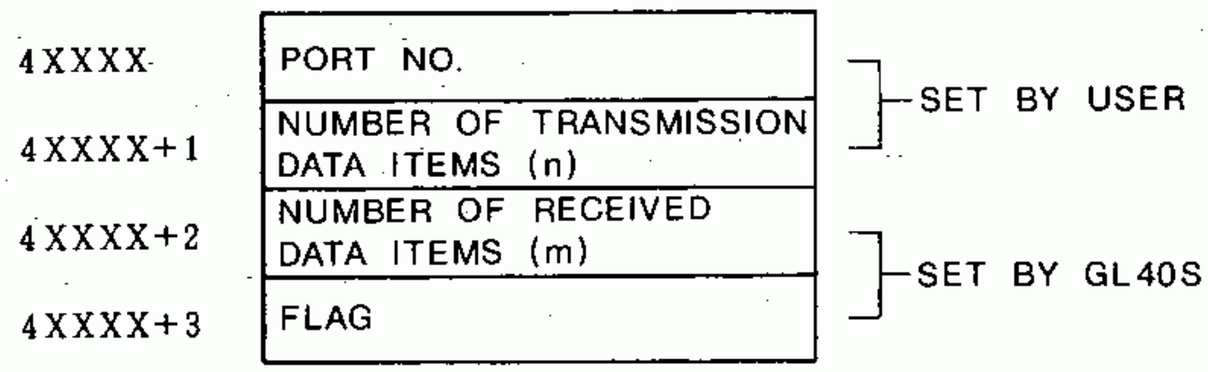
5.16.4.1 Function Block



5.16.4.2 Top Element

The top element uses four holding registers from 4XXXX, in which data to control COMM commands are stored. The four holding registers from 4XXXX can not be used in other COMM commands.

Port Nos. to be used or the number of transmission data items is specified. It is also used to indicate the number of received data items or operation condition.

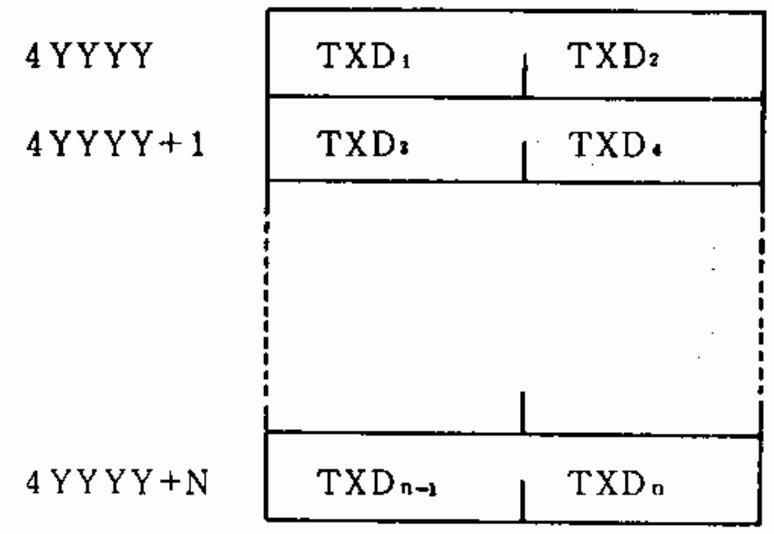


5.16.4.3 Middle Element

The middle element uses n (refer to the figures below) pieces of holding registers from 4YYYY as a transmission buffer. It indicates the head reference of register block storing transmission data. The size of the transmission buffer depends on the number n of transmission data items. Before activating command, transmission data must be set.

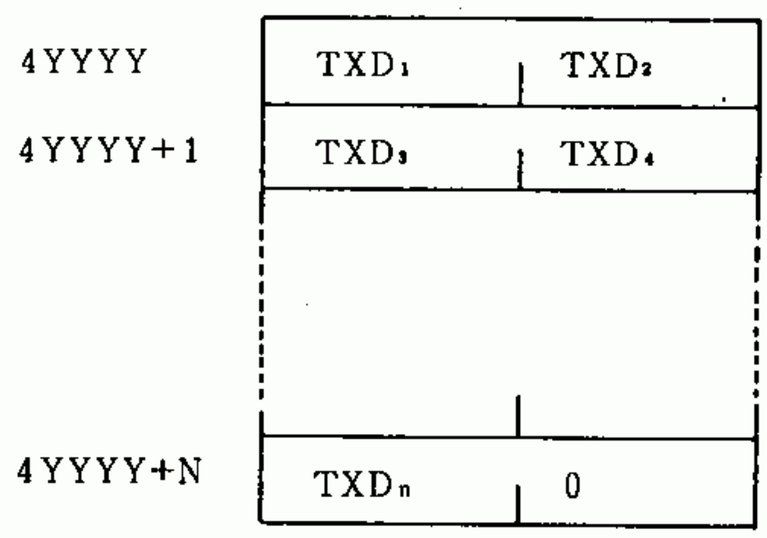
The data to be sent out is according to the small numbers beside TXD (transmission data) in the figures below.

I) n = even numbers



(N=n÷2-1)

II) n = odd numbers



(N=(n-1)÷2)

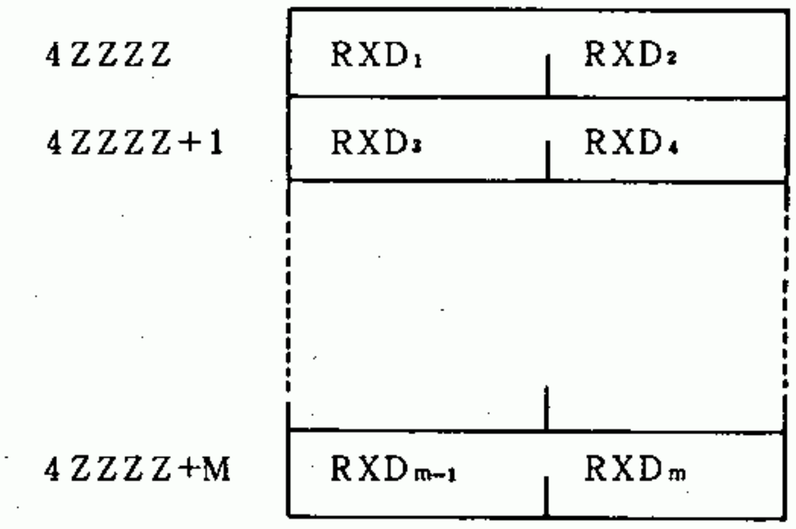
5.16.4.4 Bottom Element

The bottom element uses m (refer to the figures below) pieces of holding registers from 4ZZZZ as receiving buffer. The size of the receiving buffer depends on the number m of received data items. Therefore, the maximum number of received data items must be considered beforehand to keep the buffer. Do not use the register area for other applications since that area is kept for the receiving buffer.

This register area can not be cleared with command activation.

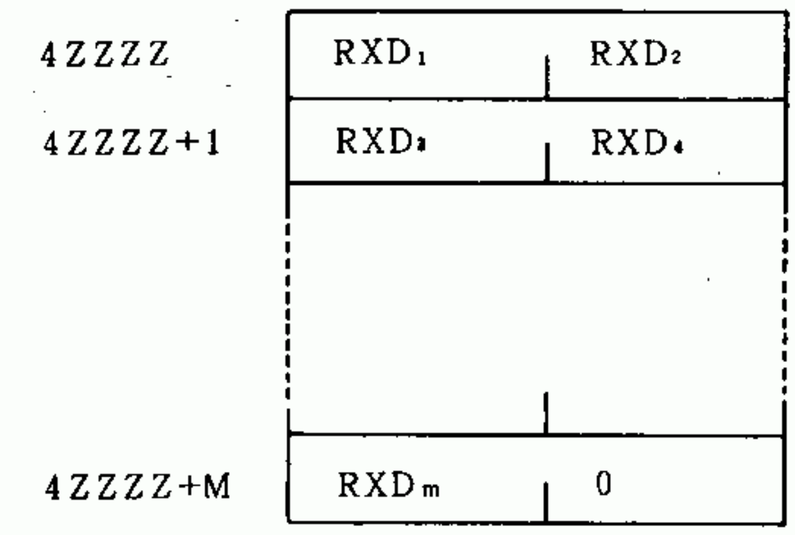
The data to be received is according to the small numbers beside TXD in the figures below.

I) m = even numbers



(M = m ÷ 2−1)

II) m = odd numbers



(M = (m − 1) ÷ 2)

5.16.4.5 Control Block

(1) Port No.(4XXXX)

Communication port of COMM is specified by the lower 4 bits of 4XXXX. Communication with external remote devices can be performed via the specified communication port and the status after execution is indicated by the upper bits 4 to 15.

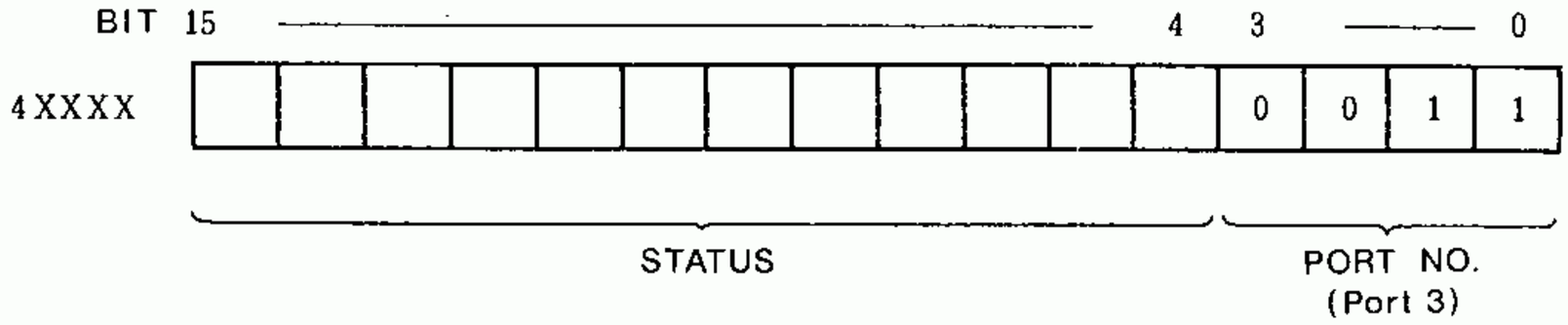


Table 5.123 Allocation of 4 × × × × Bits

Bit No.	Contents
15	BCC (CRC, LRC) error occurs only in MEMOBUS mode.
14	Transmission cannot be performed from the specified port. Hardware malfunction is possible.
13	No allocation
12	Receiving buffer overrun. The number of received data items exceeds 512 bytes.
11	The number of transmission data items is not correct. The number of 4 × × × × + 1 exceeds 512.
10	Remote devices are not connected to the specified port. DSR (data set ready) on the specified port is turned off.
9	Port No. is not correct. Port No. other than 3 to 4 is specified.
8	COMM on the specified port is defective.
7	Received response address and function code are not proper for transmisson query message. (It occurs only in MEMOBUS mode.)
6	No allocation
5*	P150 is connected to the specified port., This is when data is received from PP at command execution.
4	No allocation

Port No.		—	Bit #				Set with code.
		—	3	2	1	0	
	3	COMM, upper port	0	0	1	1	
	4	COMM, lower port	0	1	0	0	

Note Bits 4 to 15 excluding 5 are modified at every command execution.

* The error must be forced to clear before command execution when bit 5 is set. Otherwise the command will not be executed when execution command is ON. This bit is set when the port is connected with master unit such as P150. Therefore, do not connect the port for activating COMM command with P150.

(2) Number of Transmission Data Items (4XXXX+1)

The number of bytes (number of characters) of query message to be transmitted is specified. The range of n is from 1 to 512 (10).

n	Number of Transmission Buffer Registers
Even	n/2 registers
Odd	(n +1)/2 registers

When command is activated by setting a value out of the specified range, an error occurs immediately and the command is not executed.

(3) Number of Received Data Items (4XXXX+2)

The number of bytes (number of characters) of the query message to be transmitted is specified and the number is set by CUP (control panel unit) with normal completion after the command execution. The range of m value to be written in the register is from 0 to 512 (10). When the command is activated, the number of received data items is cleared to 0.

m	Number of Transmission Buffer Registers
Even	m/2 registers
Odd	(m +1)/2 registers

(4) Flag (4XXXX+3)

Command execution status is indicated.

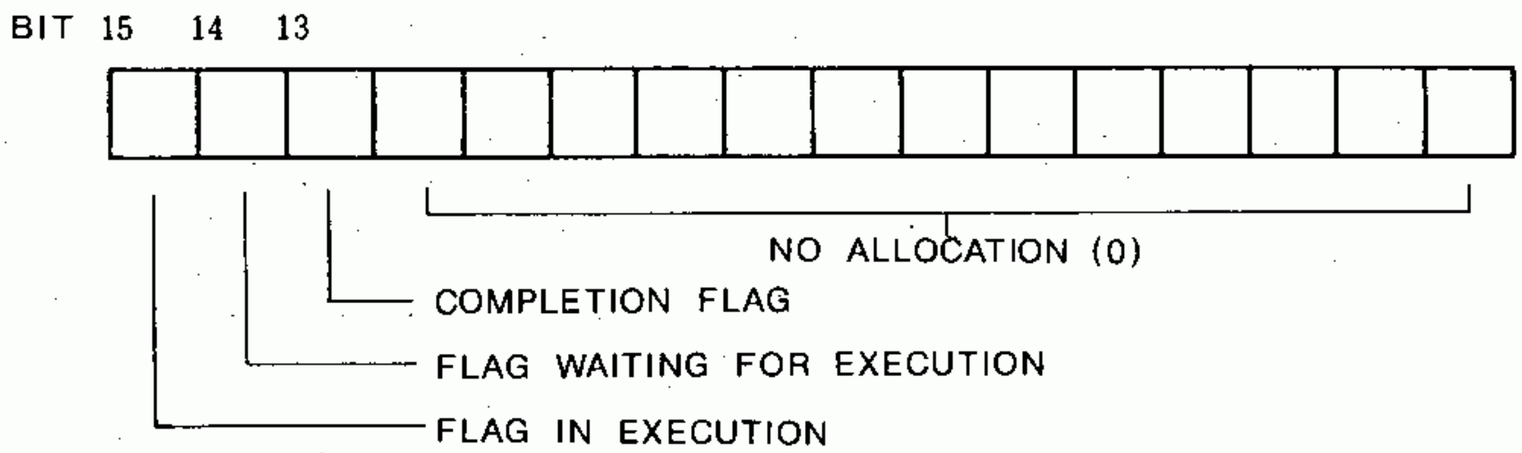


Table 5.124 Allocation of 4 × × × × + 3 Bit

Bit No.	Contents
15	This bit becomes 1 during execution. However, it does not become 1 when the execution is not activated though input 1 is accepted, since it is interlocked with output 1.
14	1 is set when the command has been activated but not executed yet. Waiting status is indicated when more than one COMM command is activated.
13	Completion of command execution is indicated. 1 is set to 1 scan by either normal completion or forced completion with error.
12 to 0	No allocation, always set at 0.

More than two bits from 13 to 15 can not be set at 1.

5.16.4.6 Definition of Input/Output

(1) Input 1: execution command

COMM command execution is instructed by input 1: ON, and input 3: OFF. Output 1 becomes ON when input 1 is accepted and the execution is started.

Note that differential contact must be used for input 1. COMM command execution is started again if this input is ON at completion of operation. Never change the transmission data during COMM command execution.

(2) Input 2: mode designation

Either transmission/receiving mode or transmission mode can be specified by input 2.

ON ....Transmission mode
OFF ...Transmission/receiving mode

In transmission mode, only transmission is performed and response from the remote unit is not required. The transmission mode is used when the remote unit is a printer, etc.

In transmission/receiving mode transmission from port is performed and then a response is waited from the remote unit. If a response is not obtained from the remote unit, waiting status will be provided and the command is completed; perform time-out process by application.

This input is checked when COMM command is activated.

(3) Input 3: command abort

COMM command during execution is aborted. When this input is ON, three inputs become OFF and have priority over other inputs.

It is used for aborting command of time-out process in transmission/receiving mode. Status error bits, excluding bit 5, are all cleared (see Table 5.123).

Note that differential contact must be used for the input in normal operation. Even if input 1 is ON, the COMM command is not executed while input 3 is ON.

* Command execution conditions
  • Input 1 goes ON from OFF.
  • Input 3 is OFF.
  • The specified port is not operating.

(4) Output 1: during execution

When input 1 is accepted and COMM command execution is started, output 1 is ON. When the operation is completed or aborted, it goes OFF.

It will be OFF when an error occurs during COMM command execution.

(5) Output 2: error

Only 1 scan is ON when COMM command execution is completed by an error. Error bits of status contains their own contents as shown in Table 5.123.

(6) Output 3: operation completion

Only 1 scan is ON when the operation is completed normally by COMM command. All status error bits become 0.

5.16.5 Application Example

5.16.5.1 Memobus Master

A message example that reads out the contents of the holding register with GL40S as MEMOBUS master unit is shown. At this time, transmission is set in MEMOBUS mode.

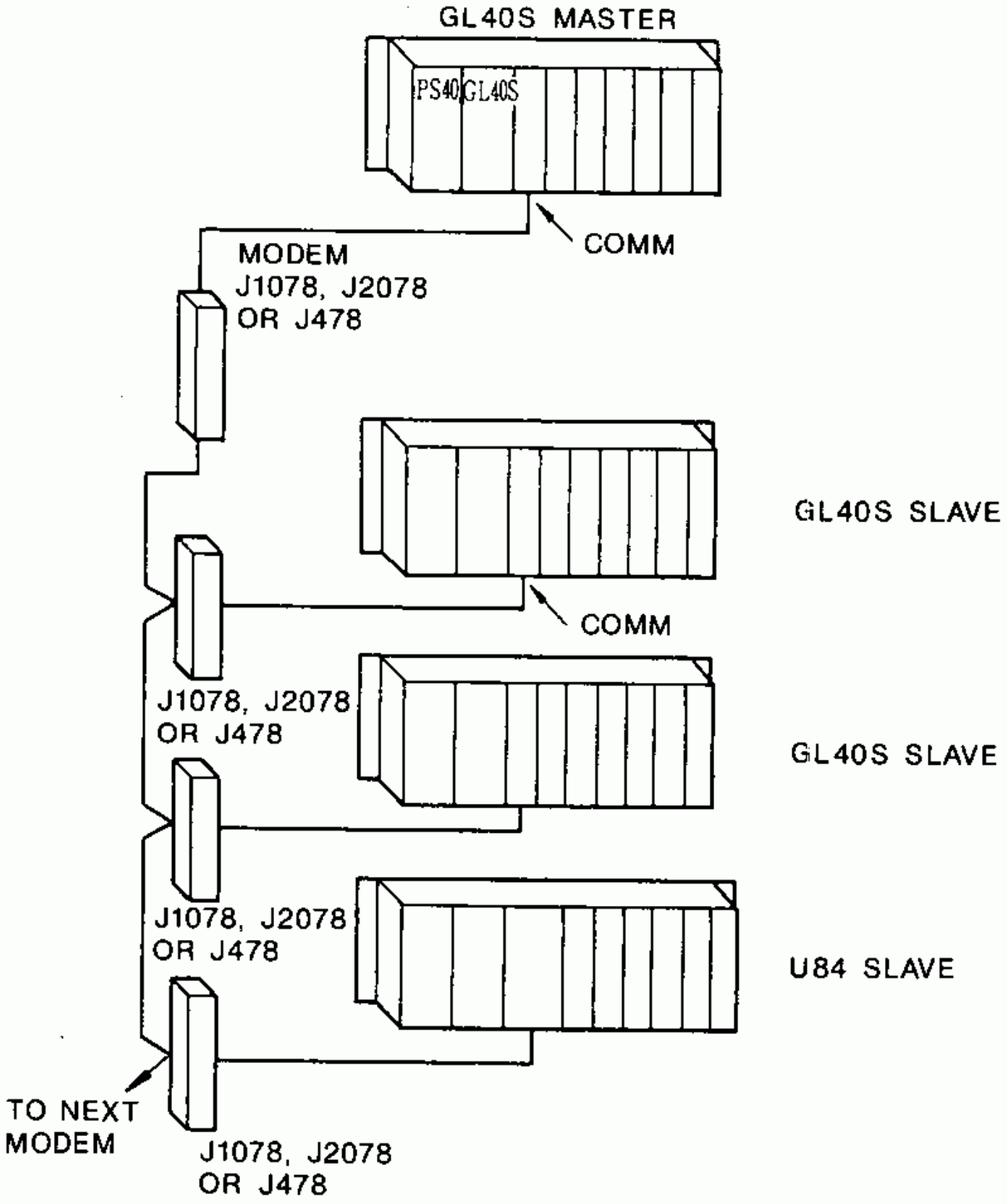


Fig. 5.99 GL40S Master Unit Configuration

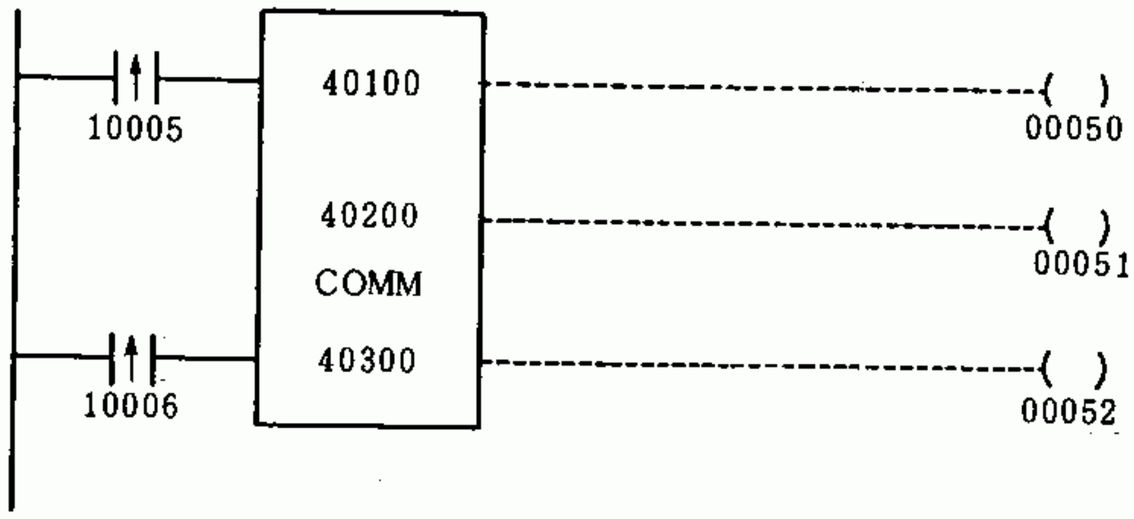


Fig. 5.100 Example of Communication Circuit

The data to be stored in each register in this example are shown below.

(1) Control Block

40100	0003 (Port No.)
40101	0006 (Number of transmission data items)

The values are in decimal.

No value is set to register 40102 for the number of received data items and register 40103 for flag. Communication is performed on the COMM upper port since the port No. is 3. The number of transmission data items is 6. This value depends on the command message to be stored in the transmission buffer.

(2) Transmission Buffer

The number of registers required as a transmission buffer is 3 since the number of transmission data items is 6. Accordingly, registers 40200 to 40202 are used as transmission buffers.

In this example, the contents of GL40S holding registers 40108 to 40110 are read out by slave address.

CRC is not needed since COMM is added automatically.

40200	03H (Slave address)	03H (Function code)	
40201	00H	6BH	··· START NO.(107)
40202	00H	03H	··· NUMBER OF HOLDING REGISTERS (3)

The values are in hexadecimal.

When the slave is U84, U84S or GL60S, register reference to be read out is any of 40108 to 40110 as GL40S. For R84H-M or GL20, it is 4108 to 4110.

(3) Receiving Buffer

Response messages from the slave are stored in registers 40300 to 40304. When there are 9 received data items, the content of register 40102 for the number of received data items becomes 9.

However, the lower 8 bits of the last register 40304 are not received data; the number becomes 0.

40300	03H	03H
40301	06H	02H
40302	2BH	00H
40303	00H	00H
40304	63H	00H

The values are in hexadecimal.

(4) Operation

① 10005 goes OFF from ON. At this time, coil 50 is ON and communication to the specified slave unit are performed unless the COMM command to the same port is not operating.
② One scan of coil 52 is ON when response message receiving is completed.

5.16.5.2 Transmission to ASCII Devices

Data transmission to ASCII devices (a printer in this example) is described here. Suppose the character lines in Fig. 5.102 are to be printed.

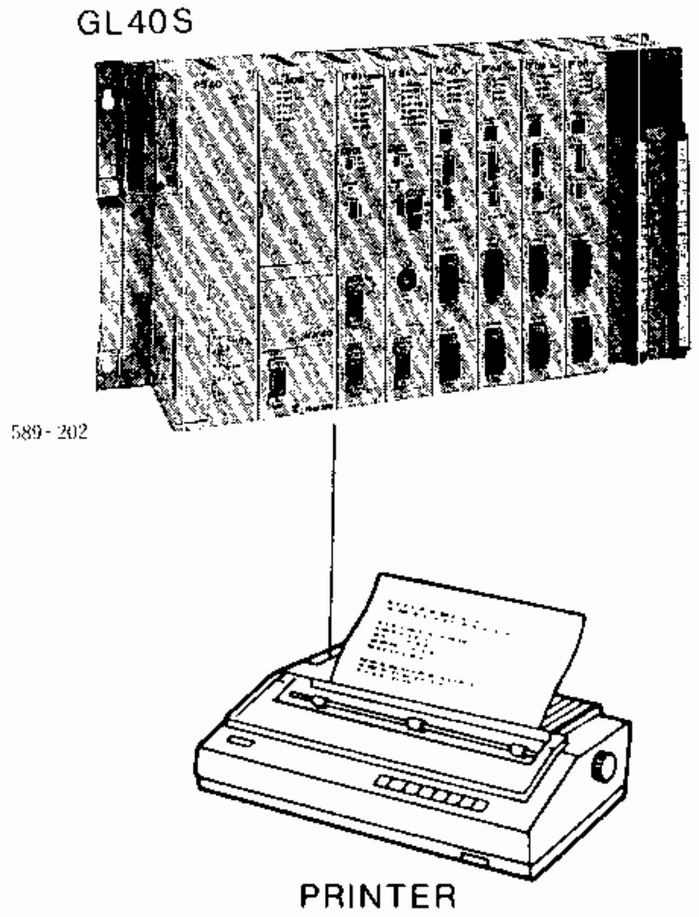


Fig. 5.101 Connection with Printer

```
bbbbbbPRODUCTIONbREPORT
bbbbbbDATE : b03b12bTIME : b10b46
bbbbbbMACHINEbNO.GL40S
```

(b : space)

Fig. 5.102 Example of Printing

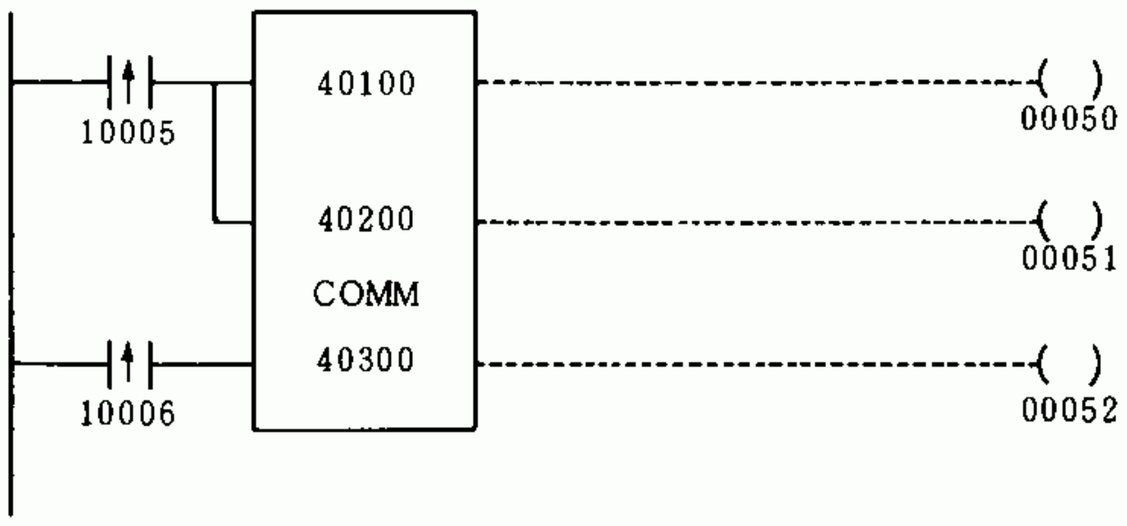


Fig. 5.103 Example of Communication Circuit

The following shows data to be stored in each register in this example.

(1) Control Block

40100	0003 (Port No.)
40101	0077 (Number of transmission data items)

The values are in decimal.

No value is set to register 40103 for flag and register 40102. The number of transmission data items includes "CR" and "LF" in each line.

(2) Transmission Buffer

The number of registers required as a transmission buffer is 39 since the number of transmission data items is 77. Accordingly, registers 40200 to 40238 are used as transmission buffers.

40200	20H	20H
40201	20H	20H
40202	20H	50H (P)
40203	52H (R)	4FH (O)
40204	44H (D)	55H (U)
40205	43H (C)	54H (T)
40206	49H (I)	4FH (O)
40207	4EH (N)	20H
40208	52H (R)	45H (E)
40209	50H (P)	4FH (O)
40210	52H (R)	54H (T)
40211	0DH (CR)	0AH (LF)
40212	20H	20H
40213	20H	20H
40214	20H	44H (D)
40215	41H (A)	54H (T)
40216	45H (E)	3AH (:)
40217	20H	30H (0)
40218	33H (3)	20H
40219	31H (1)	32H (2)
40220	20H	54H (T)
40221	49H (I)	4DH (M)
40222	45H (E)	3AH (:)
40223	20H	31H (1)
40224	30H (0)	20H
40225	34H (4)	36H (6)
40226	0DH (CR)	0AH (LF)
40227	20H	20H
40228	20H	20H
40229	20H	4DH (M)
40230	41H (A)	43H (C)
40231	48H (H)	49H (I)
40232	4EH (N)	45H (E)
40233	20H	4EH (N)
40234	4FH (O)	2EH (.)
40235	20H	47H (G)
40236	4CH (L)	36H (6)
40237	30H (0)	53H (S)
40238	0DH (CR)	0AH (LF)

The values are in hexadecimal.

(3) Receiving Buffer

There is no received data. Therefore, a receiving buffer is not necessary in this case. However, in COMM command, temporary designation is performed in the bottom element. Register 40300 in this example is not used in actual COMM command.

(4) Operation

① 10005 goes OFF from ON. At this time, coil 50 is ON and the printing in Fig. 5.102 is performed unless COMM command to the same port is not operating.
② One scan of coil 52 is ON when printing is completed.

5.16.6 Precautions for Use

For communication by using COMM command, follow the precautions below.

① In transmission/receiving mode, response waiting status is provided when no response returns from the remote unit. The waiting status can be released by input 3.

② After command activation, if any of the following occurs, the COMM command execution is aborted.
- Transmission data are changed.
- Register contents of control block are changed.
- Network including activated COMM command are skipped.
- Activated COMM command is changed from P150.

③ When input goes ON from OFF, the command execution is started. However, if the port has COMM command in execution, a command executed later is in the waiting status and then executed when the former COMM command is completed.

④ Control input for command activation is by differential signals.

⑤ When data required in the specified COMM command are set to registers by arithmetic operation command, etc., it must be performed with differential signals before COMM command is activated. If using level signals, trouble, such as change of unexpected register (contents) not prepared in item ② above may occur.

⑥ When data are sent to the slave unit by simultaneous MEMOBUS broadcasting, communication is performed in the transmission mode. At this time, do not activate more than one COMM command simultaneously to the port. The slave side cannot receive the data properly since the MEMOBUS command interval becomes shorter than the specified value.

⑦ When more than one COMM command is activated and one is aborted during execution, the same error as ⑥ may occur.

⑧ Do not connect any master unit such as P150 with the port used as the master port; the port may hang up. Should this happen, stop the COMM command activating the port by using RAP, etc.

5.16.7 Cables

Connect COMM and the remote unit after carefully checking the connection specifications of the remote unit. The following are cables meeting YASKAWA standards. Other cables to be used are supplied by the customers.

(1) SC Connection Cables

Connection cables for communication with GL40S, U84, GL60S or U84S are as follows.

Note In Tables 5.125 to 5.131, the meanings of abbreviations are as follows:

1	PGND	protection ground	5	CTS	clear to send
2	TXD	transmission data	6	DSR	data set ready
3	RXD	received data	7	SGND	signal ground
4	RTS	request to send	8	CD	carrier detection
			9, 20	DTR	data terminal ready

Cable Type	Length
JZMSZ-W1019-1	5m
JZMSZ-W1019-2	15m

Table 5.125 Connection of JZMSZ-W1019

Pin No.	Signal Name	Direction	Pin No.	Signal Name	Color
1	PGND	⟷	1	PGND	Brown
2	TXD	→	3	RXD	Orange/red
3	RXD	←	2	TXD	Red/orange
4	RTS	⌐	4	RTS	
5	CTS	⌙	5	CTS	
6	DSR	→	9	DTR	White/blue
9	DTR	←	6	DSR	Blue/white
7	SGND	⟷	7	SGND	Black

(2) ASCII Device Connection Cables

Standard cable JZMSZ-W1018 is provided for connection with ASCII devices.

Cable Type	Length
JZMSZ-W1018-1	5m
JZMSZ-W1018-2	15m

Table 5.126 Connection of JZMSZ-W1018

COMM Side		Direction	ASCII Device Side		Color
Pin No.	Signal Name		Pin No.	Signal Name	
1	PGND	⟷	1	PGND	Brown
2	TXD	→	3	RXD	Red
3	RXD	←	2	TXD	Orange
		⟶	4	RTS	
		⟶	5	CTS	
5	CTS	⟷	6	DSR	Green
		⟶	20	DTR	
6	DSR	⟵			
9	DTR	⟵			
7	SGND	⟷	7	SGND	Black

(3) Modem Connection Cables

When the connection distance of the communication circuit exceeds 15 m, or when more than one slave unit is used in MEMOBUS system, modem is required.

Table 5.127 Modem Connection Cables

Cable Types JZMSZ-	Application	Modem	Length
W1007-1	For ASCII device	DISCT-J478	5m
W1007-T1	For ASCII device	DISCT-J1078	5m
W1008-1	For ASCII device	DISCT-J478	5m
W1008-T1	For ASCII device	DISCT-J1078	5m
W1017-1	For MEMOBUS	DISCT-J478	5m
W1017-2	For MEMOBUS	DISCT-J478	15m
W1017-T1	For MEMOBUS	DISCT-J1078	5m
W1017-T2	For MEMOBUS	DISCT-J1078	15m
W2020-1	For MEMOBUS	DISCT-J2078	2.5m
W2020-2	For MEMOBUS	DISCT-J2078	5m
W2020-3	For MEMOBUS	DISCT-J2078	10m
W2020-4	For MEMOBUS	DISCT-J2078	15m

Table 5.128 JZMSZ-W1007 Connection

COMM Side		Direction	Modem Side		Color
Pin No.	Signal Name		Pin No.	Signal Name	
1	PGND	⟷	1	PGND	Brown
2	TXD	→	2	TXD	Red
3	RXD	←	3	RXD	Orange
4	RTS	↱	4	RTS	Yellow
9	DTR	→	20	DTR	White
5	CTS	←	8	CD	Green
6	DSR	←	6	DSR	Blue
7	SGND	⟷	7	SGND	Black

5.16.7 Cables (Cont'd)

Table 5.129 JZMSZ-W1008 Connection

Modem Side		Direction	ASCII Device Side		Color
Pin No.	Signal Name		Pin No.	Signal Name	
1	PGND	⟷	1	PGND	
2	TXD	→	2	TXD	Red
3	RXD	←	3	RXD	Orange
		┐	4	RTS	
		┘	5	CTS	
4	RTS	←			
20	DTR	←	20	DTR	Brown
6	DSR	→	6	DSR	Blue
7	SGND	⟷	7	SGND	White

Table 5.130 JZMSZ-W1017 Connection

COMM Side		Direction	Modem Side		Color
Pin No.	Signal Name		Pin No.	Signal Name	
1	PGND	⟷	1	PGND	Brown
2	TXD	→	2	TXD	Red
3	RXD	←	3	RXD	Orange
4	RTS	→	4	RTS	Yellow
5	CTS	←	5	CTS	Green
6	DSR	←	6	DSR	Blue
9	DTR	→	20	DTR	White
7	SGND	⟷	7	SGND	Black

Table 5.131 JZMSZ-W2020 Connection

COMM Side		Direction	Modem Side		Color
Pin No.	Signal Name		Pin No.	Signal Name	
1	PGND	⟷	1	PGND	Brown
2	TXD	→	2	TXD	Red
3	RXD	←	3	RXD	Orange
4	RTS	→	4	RTS	Yellow
5	CTS	←	5	CTS	Green
6	DSR	←	6	DSR	Blue
9	DTR	→	9	DTR	White
7	SGND	⟷	7	SGND	Black

(4) Modem-to-modem Connection Cables

For modem-to-modem (DISCT-J478, DISCT-J1078, DISCT-J2078) connection, two-core twisted cables (RG-108/U or equivalent) must be used.

5.17 MOTION COMMANDS

Motion commands listed in Table 5.132 execute motion control, used in ladder diagrams. They cannot start multiple commands given simultaneously to the same axis. The monitor, however, can start with another command simultaneously.

For details, refer to the "Memocon-SC GL40S User's Manual Motion Control Version (SIE-C815-15.30)".

Table 5.132 List of Motion Command Functions

No.	Name	Symbol	Functions
1	1-axis program operation	MOVI	Executes 1-axis program operation from a designated block
2	1-axis single block operation	MVIS	Executes 1-axis operation of only a designated block
3	Independent multi-axis program operation	MOVM	Executes multi-axis simultaneous-start operation from a designated block
4	3-axis program operation	MOVL	Executes 2- or 3-axis program operation from a designated block
5	3-axis single block operation	MVLS	Execute 2- or 3-axis operation of only a designated block
6	1-axis zero point return operation	ZRN	Executes 1-axis zero point return operation
7	JOG operation	JOG	Executes jog operation
8	Handle operation	HNDL	Switches the operation mode to the handle operation mode
9	Monitor	MON	Monitors alarms, parameters, program No., etc
10	Current position setting	POS	Changes current position data
11	Parameter setting	PRM	Sets a parameter
12	Variable setting	VAR	Sets values of variables H1, H2, H3 and H4.
13	Alarm reset	ARES	After the occurrence of an alarm, resets the alarm
14	Servo ON	SVON	Turns ON power to the motor Turns OFF power to the motor
15	Current position adjustment	ADJ	Adjusts current positions before executing 2- or 3-axis operations

5.17.1 Motion Command Functions

The motion control of GL40S has the following motion functions. Acceleration and deceleration curves of each operation are determined by setting parameters.

(1) Memory Motion Functions (Automatic Operation Group)

- 1-axis Program Operation (MOVI)
- 1-axis Single Block Operation (MVIS)
- Independent Multi-axis Progam Operation (MOVM)
- 3-axis Program Operation (MOVL)
- 3-axis Single-Block Operation (MVLS)

(MOVL) and (MVLS) → Current Position Adjustment (ADJ)

These commands are used for executing automatic operation by use of motion programs stored in the Servopack memory.

(2) Jog Operation Function – JOG Operation (JOG)

This command is used for feeding the machine manually. Feed speed is selected out of 4 speeds represented by parameters (P04, P31, P32, P33).

(3) Zero Point Return Operation Function – 1–axis Zero Point Return Operation (ZRN)

This operation is to return the machine to its inherent zero point, using a PG with zero point pulse and an external limit smitch indicating the zero point area.

Servopack, combined with a motor equipped with an absolute encoder, does not require zero point return operation if the machine zero point is once set on the front panel of Servopack .

(4) Handle Operation Function–Handle Operation (HANDL)

This command is used for handle operation. Turning the handle (manual pulse) connected to Servopack during ON i.e. execution of HANDL moves the machine in the positive or negative direction according to handle turning directions.

(5) Setting and Monitor
- Monitor (MON)
- Current Position Setting (POS)
- Parameter Setting (PRM)
- Variable Setting (VAR)
- Current Position Adjustment (ADJ)

POS is used for changing the machine current position; PRM for changing parameters in Servopack; VAR for setting variables and constants in motion programs; and ADJ for adjusting current positions among 2- or 3-axis interpolating operations.

The MON command is used for monitoring the Servopack status when needed.

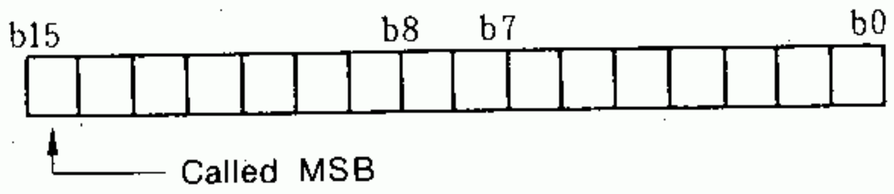
(6) Servo Control Functions
- Servo ON (SVON)
- Alarm Reset (ARES)

SVON is used for controlling the Servopack main circuits to turn Power to the motor ON or OFF. ARES is used, after Servopack detection of an alarm, for resetting the alarm.

5.17.2 Description of Motion Commands

Symbols such as 4XXXX used in this description indicate holding register numbers in GL40S.

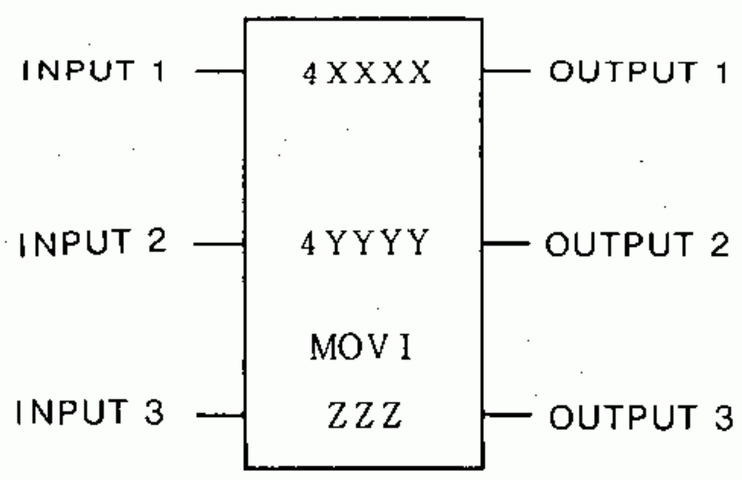
Holding registers enable the setting of ± numeric data and the input of monitor data. A holding register consists of 16 bits, and it takes a negative value when 1 is set to its MSB.



MSB (b15) = 0: Means positive value.

MSB (b15) = 1: Means negative value.

5.17.2.1 1-Axis Program Operation (MOVI)



Entering the START command will start operation from a block having designated O# and N# and execute program contents in regular sequence until the program end command (M30) appears.

Before entering the START command, Servopack must be loaded with motion programs for one axis.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

4XXXX	Module No.
4XXXX+1	Axis No.
4XXXX+2	O No.
4XXXX+3	N No.
4XXXX+4	FEEDHOLD Selection
4XXXX+5	System use

IF66 No. : 1 to 4

Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)

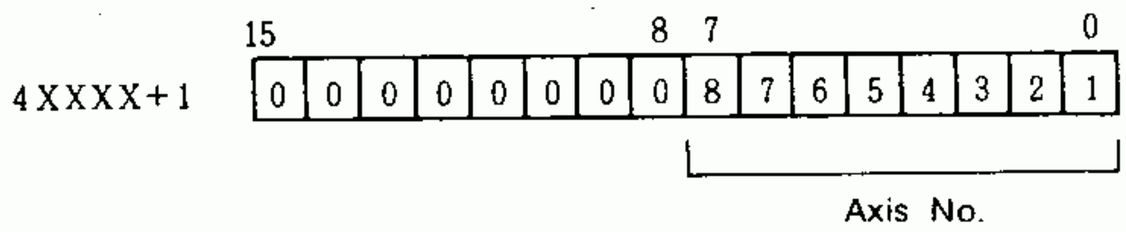
Motion program O# : 1 to 20 (DEC)

Motion program N# : 1 to 999 (DEC)

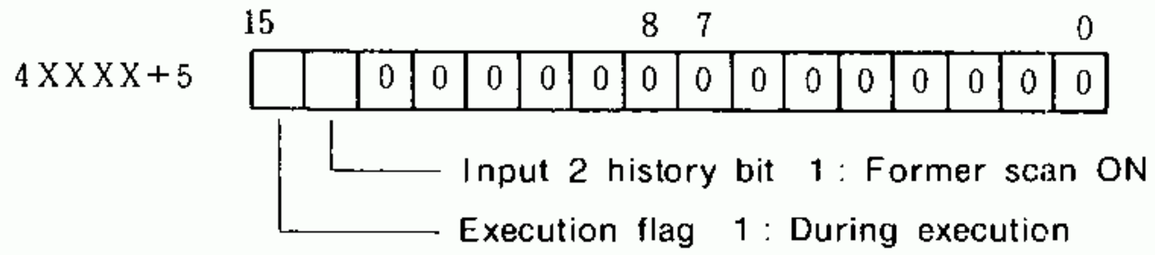
1 when Input 2 is used as FEEDHOLD command.

Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.

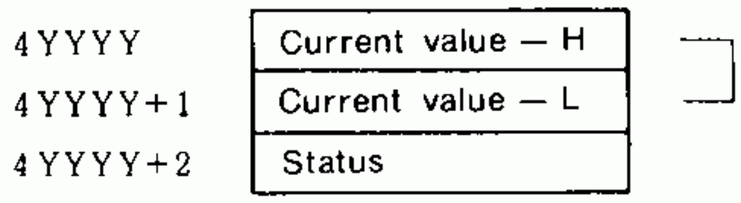


*2 System use (Do not change on the ladder.)



Data Monitored

Current values and error status are displayed.



0 to ± 9 9 9 9 9 9 9 9 (DEC)

4YYYY	4YYYY+1

It is displayed if an error occurs.

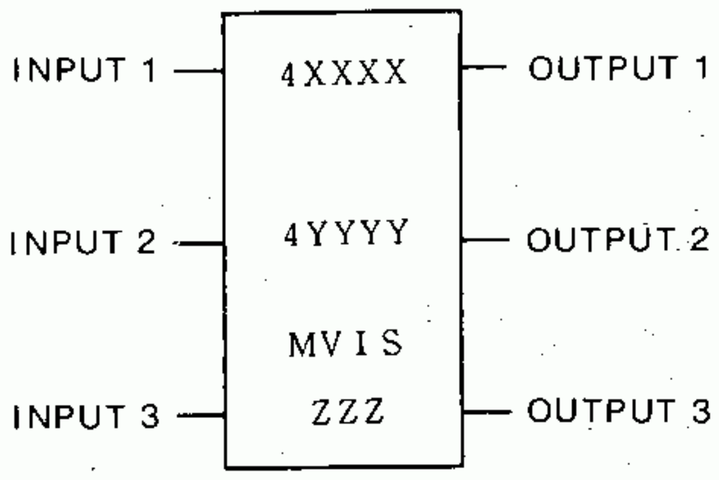
Refer to Par.5.17.2.16.

When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. If Input 3 is ON before, START is unacceptable.
Input 2	When FEEDHOLD selection flag ≠ 1, continue command : MFIN MFIN signal is output with rise differentiation (⊣↑⊢). However, this is effective only when MFIN is required (M50 : during execution). When FEEDHOLD selection flag = 1, hold command : FEEDHOLD While FEEDHOLD is ON, execution is temporarily stopped. When it turns OFF, the remaining program is executed. FEEDHOLD is acceptable only when Output 1 is ON.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN Run is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : M30 When execution is ended without an error, only 1 scan turns ON.

5.17.2.2 1-Axis Single Block Operation (MVIS)

This operation is to execute a motion program of an axis block-by-block.



Entering the START command will execute the operation of only a block having designated O# and N#.

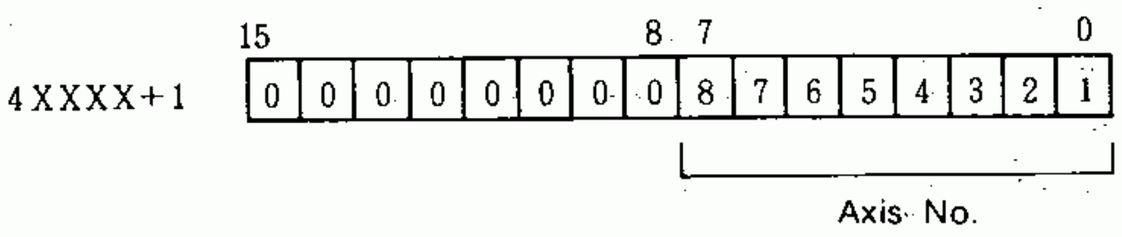
Before entering the START command, Servopack must be loaded with motion programs for one axis.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

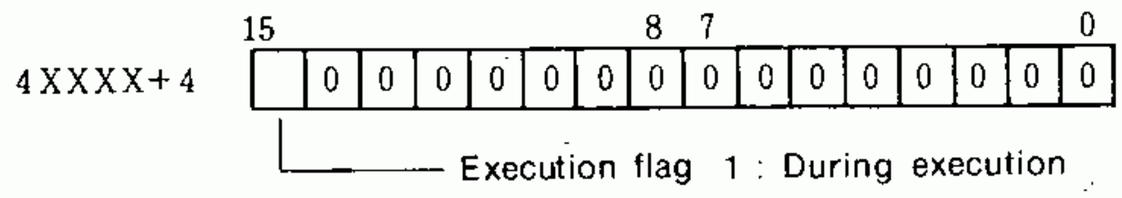
4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	O No.	Motion program O# : 1 to 20 (DEC)
4XXXX+3	N No.	Motion program N# : 1 to 999 (DEC) (*2)
4XXXX+4	FEEDHOLD	
4XXXX+5	System use	Used as execution flag in systems (* 3)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit. Simultaneous designation of multiple axes is impossible.



*2 When execution is ended without an error, N# becomes (N + 1)# automatically. When M30 is executed in block operation, N# becomes 1.

*3 System use (Do not change on the ladder.)



Data Monitored

Current values and error status are displayed.

4YYYY	Current value — H
4YYYY+1	Current value — L
4YYYY+2	Status

0 to ± 9 9 9 9 9 9 9 9 (DEC)
4YYYY 4YYYY+1

It is displayed if an error occurs.

Refer to Par.5.17.2.16.

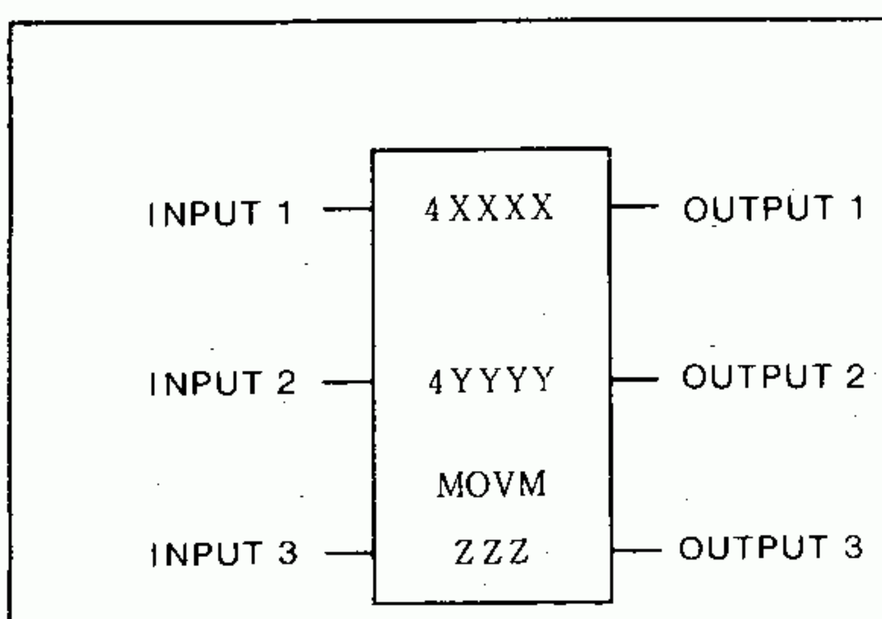
When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. If Input 3 is ON before, START is unacceptable.
Input 2	When FEEDHOLD selection flag ≠ 1, continue command : MFIN MFIN signal is output with rise differentiation (⊣↑⊢). However, this is effective only when MFIN is required (M50 during execution). When FEEDHOLD selection flag = 1, hold command : FEEDHOLD While FEEDHOLD is ON, execution is temporarily stopped. When it turns OFF, the remaining program is executed. FEEDHOLD is acceptable only when Output 1 is ON.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN Run is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : M30 When execution is ended without an error, only 1 scan turns ON.

5.17.2.3 Independent Multi-axis Program Operation (MOVM)

MOVM operation is used when Servopack axes are desired to start simultaneously. All axes, starting simultaneously, continue operation so far as to reach M30 (execution end) of their respective programs.

Since the program number (O#) and sequence block number (N#) are common data to all axes, the axes designated by MOVM must have motion programs that have the same O# and also the same N#. If an axis lacks a motion program, the axes that started operation enter stop motion.



Entering the START command will start the operation of designated axes from a block having designated O# and N# and execute program contents until the program end commands (M30) of all axes appear.

Before entering the START command, Servopack must be loaded with motion programs for all designated axes.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

4XXXX	Module No.
4XXXX+1	Axis No.
4XXXX+2	O No.
4XXXX+3	N No.
4XXXX+4	Axis No. for MFIN
4XXXX+5	FEEDHOLD Selection
4XXXX+6	System use

IF66 No. : 1 to 4

Axis No. of Servopack : Set 1 to any bits of low-order byte (*1)

Motion program O# : 1 to 20 (DEC)

Motion program N# : 1 to 999 (DEC)

Axis that outputs MFIN : Set 1 to any bits of low-order byte (*1)

1 when Input 2 is used as FEEDHOLD command.

Used as execution flag in systems (*2)

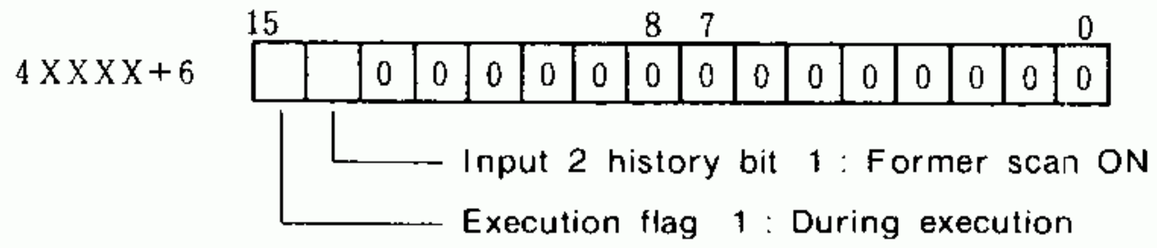
Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.

4XXXX+1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1

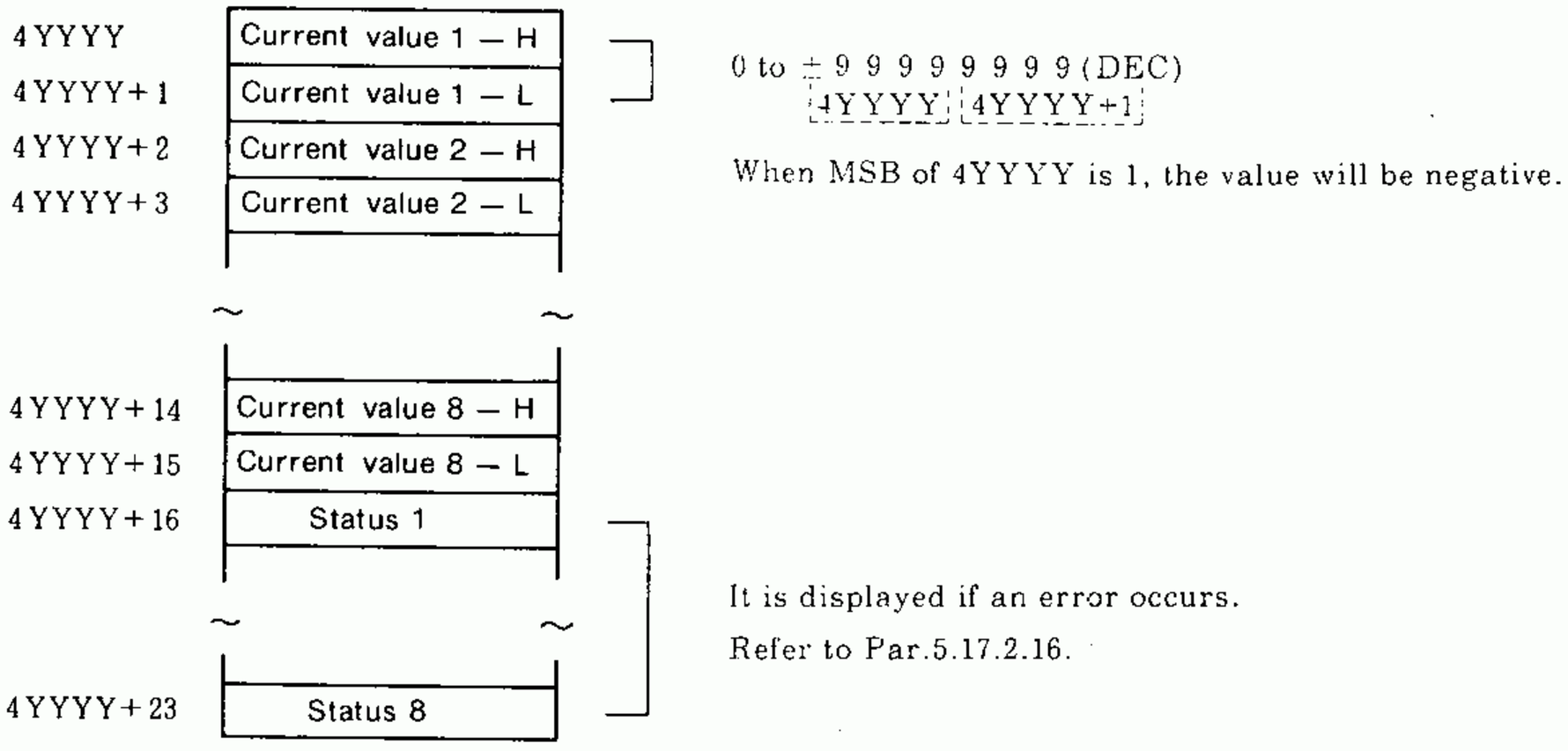
Axis No.

*2 System use (Do not change on the ladder.)



Data Monitored

Current values and error status are displayed from the smallest number of designated axes.



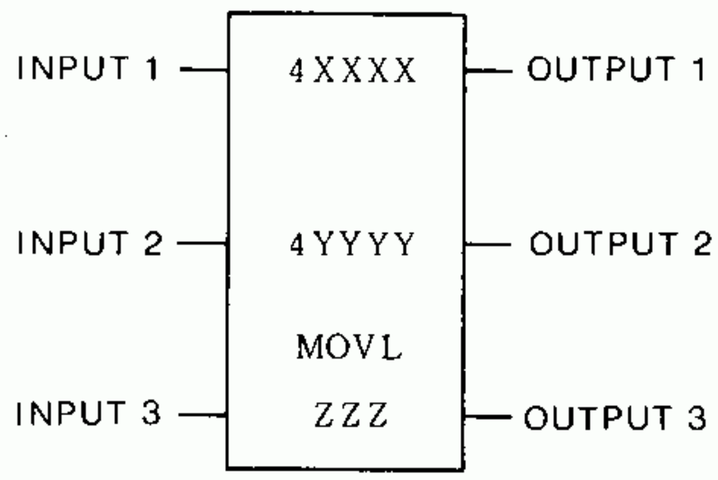
Note 24 holding register are kept for monitor data regardless of number of axes. In the case of less than 8 axes, current value is displayed from 4YYYY and status from 4YYYY + 16.

MOVM

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. If Input 3 is ON before, START is unacceptable.
Input 2	When FEEDHOLD selection flag ≠ 1, continue command : MFIN MFIN signal is output with rise differentiation (⊣↑⊢). However, this is effective only when MFIN is required (M50 : during execution). When FEEDHOLD selection flag = 1, hold command : FEEDHOLD While FEEDHOLD is ON, execution is temporarily stopped. When it turns OFF, the remaining program is executed. FEEDHOLD is acceptable only when Output 1 is ON.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN Run is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : M30 When execution is ended without an error, only 1 scan turns ON.

5.17.2.4 3-Axis (2-Axis) Program Operation (MOVL)

MOVL operation is used when 2-or 3-axis interpolation motion is desired. Execute MOVL, after loading Servopack with the same motion program for designated interpolation axes and executing the current position adjustment (ADJ) command.



Entering the START command will start the operation of 2 or 3axes from a block having designated O# and N# and execute respective program contents until the program end command (M30) appears.

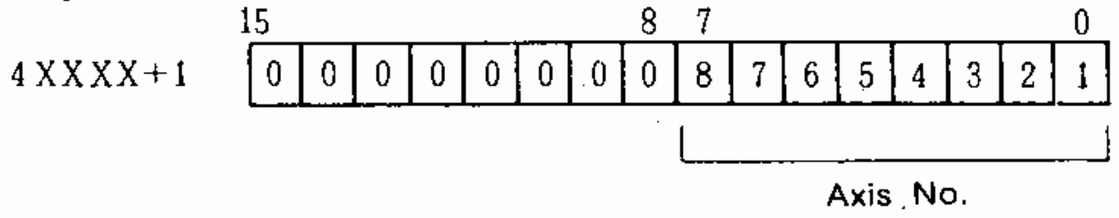
Before entering the START command, Servopack must be loaded with motion program for 2 or 3 axes.

Note Set ZZZ to any value from 1 to 999 (DEC).

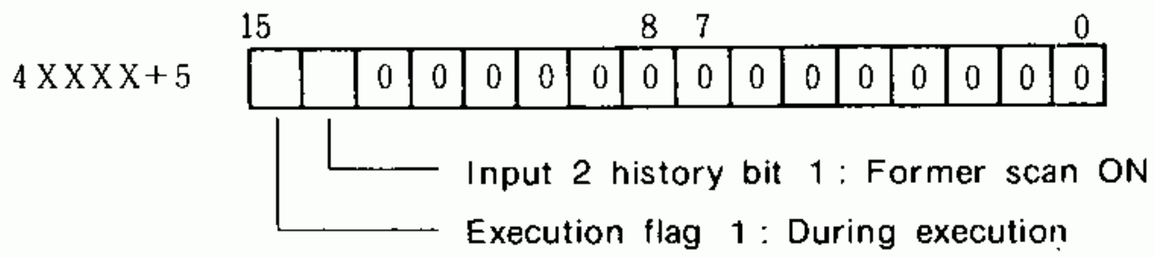
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to 2 or 3 bits of the low-order byte (*1)
4XXXX+2	O No.	Motion program O# : 1 to 20 (DEC)
4XXXX+3	N No.	Motion program N# : 1 to 999 (DEC)
4XXXX+4	FEEDHOLD Selection	1 when Input 2 is used as FEEDHOLD command.
4XXXX+5	System use	Used as execution flag in systems (*2)

Note ***1 Set 1 to 2 or 3 axes to be designated from 0 bit to 7 bit.**



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

Note Execute MOVL, after executing ADJ (current position adjustment) for axes to be designated.

Data Monitored

Current values and error status are displayed from the smallest number of designated axes.

4YYYY	Current value 1 — H
4YYYY+1	Current value 1 — L
4YYYY+2	Current value 2 — H
4YYYY+3	Current value 2 — L
4YYYY+4	Current value 3 — H
4YYYY+5	Current value 3 — L
4YYYY+6	Status 1
4YYYY+7	Status 2
4YYYY+8	Status 3

0 to ± 9 9 9 9 9 9 9 9 (DEC)
[4YYYY] [4YYYY+1]

When MSB of 4YYYY is 1, the value will be negative.

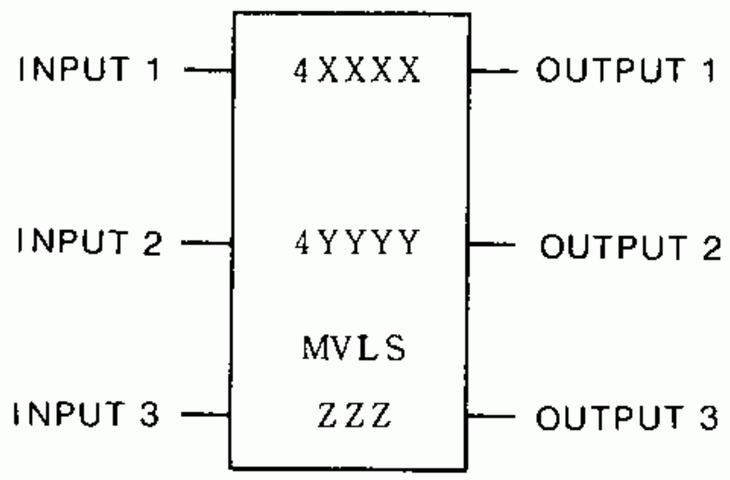
It is displayed if an error occurs.

Refer to Par.5.17.2.16.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. If Input 3 is ON before, START is unacceptable.
Input 2	When FEEDHOLD selection flag ≠ 1, continue command : MFIN MFIN signal is output with rise differentiation (⊣↑⊢). However, this is effective only when MFIN is required (M50 : during execution). When FEEDHOLD selection flag = 1, hold command : FEEDHOLD While FEEDHOLD is ON, execution is temporarily stopped. When it turns OFF, the remaining program is executed. FEEDHOLD is acceptable only when Output 1 is ON.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN Run is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : M30 When execution is ended without an error, only 1 scan turns ON.

5.17.2.5 3-Axis (2-Axis) Single-Block Operation (MVLS)

This operation is to execute motion programs for 2 or 3 axes block-by-block. The current position adjustment (ADJ) command must be executed for designated axes to adjust positions or 2 of 3 axes. When block-by-block operation is executed successively, start the ADJ command once before executing MVLS.



Entering the START command will execute 2- or 3-axis operation of only a block having designated O# and N#.

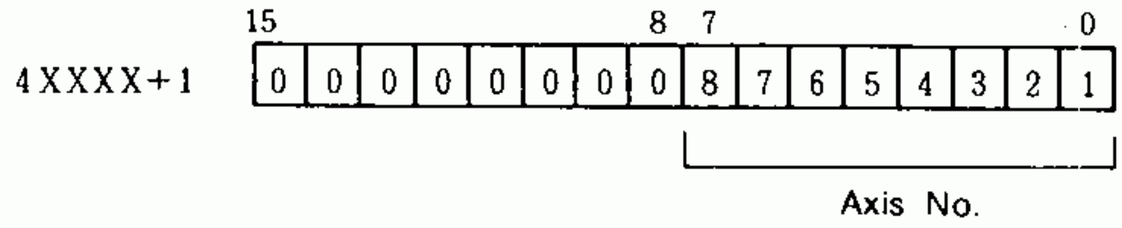
Before entering the START command, Servopack must be loaded with motion program for 2 or 3 axes.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

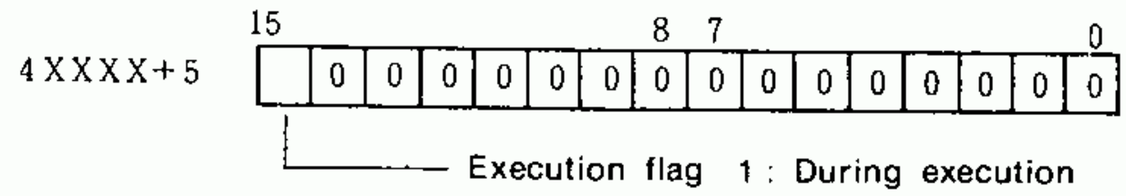
4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to 2 or 3 bits of the low-order byte (*1)
4XXXX+2	O No.	Motion program O# : 1 to 20 (DEC)
4XXXX+3	N No.	Motion program N# : 1 to 999 (DEC) (*2)
4XXXX+4	FEEDHOLD	
4XXXX+5	System use	Used as execution flag in systems (*3)

Note *1 Set 1 to 2 or 3 axes to be designated from 0 bit to 7 bit.



*2 When execution is ended without an error, N# becomes (N + 1)# automatically. When M30 is executed in block operation, N# becomes 1.

*3 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

MVLS

Data Monitored

Current values and error status are displayed from the smallest number of designated axes.

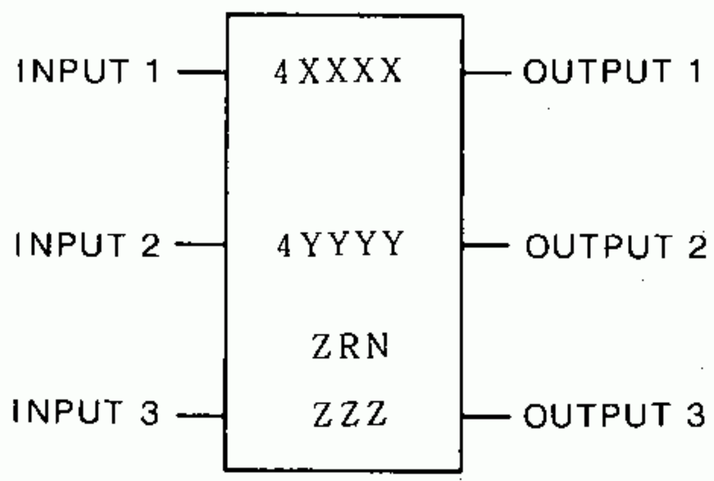
4YYYY	Current value 1 — H	0 to ± 9 9 9 9 9 9 9 9 (DEC) 4YYYY 4YYYY+1 When MSB of 4YYYY is 1, the value will be negative.
4YYYY+1	Current value 1 — L	
4YYYY+2	Current value 2 — H	
4YYYY+3	Current value 2 — L	
4YYYY+4	Current value 3 — H	
4YYYY+5	Current value 3 — L	
4YYYY+6	Status 1	It is displayed if an error occurs. Refer to Par.5.17.2.16.
4YYYY+7	Status 2	
4YYYY+8	Status 3	

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. If Input 3 is ON before, START is unacceptable.
Input 2	Hold command: FEEDHOLD While FEEDHOLD is ON, execution is temporarily stopped. When it turns OFF, the remaining program is executed. FEEDHOLD is acceptable only when Output 1 is ON.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN Run is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : END OF BLOCK OPERATION When execution is ended without an error, only 1 scan turns ON.

5.17.2.6 1-Axis Zero Point Return Operation (ZRN)

To execute zero point operation, set parameters beforehand so as to obtain P18 - b4 = 1 to select the use of zero point return operation.

Operation patterns of zero point return operation include mode I and mode II. Set parameters beforehand to select either mode.



Entering the START command will execute 1-axis zero point return operation in accordance with set parameters(*1).

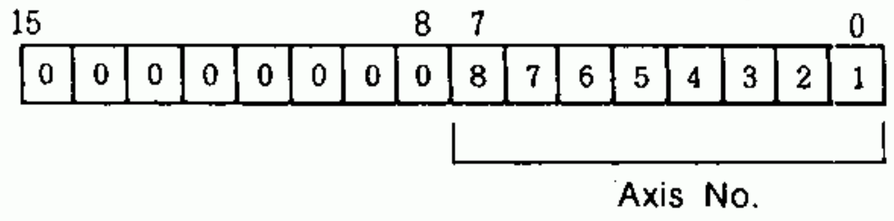
Note Set the following parameters.
Select Zero point return → Used on the DETAIL screen.
Depress the CONTENTS label and set details on the small screen.

Note Set ZZZ to any value from 1 to 999 (DEC).

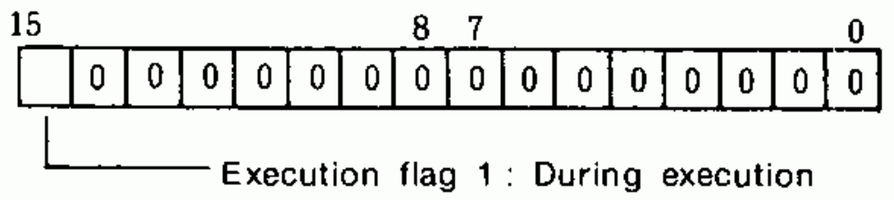
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

ZRN

Data Monitored

Current values and error status are displayed.

4YYYY	Current value — H
4YYYY+1	Current value — L
4YYYY+2	Status

0 to ± 9 9 9 9 9 9 9 9 (DEC)
[4YYYY] [4YYYY+1]

It is displayed if an error occurs.

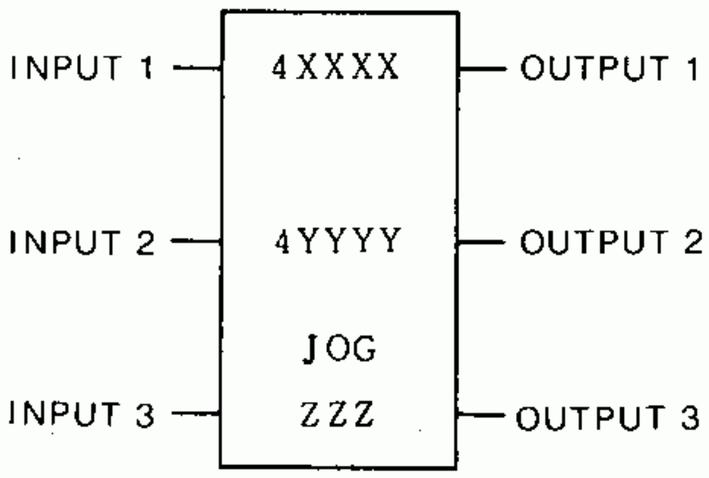
Refer to Par. 5.17.2.16.

When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Reverse rotation command : REVERSE REVERSE designates the direction of rotation : OFF for forward running, and ON for reverse running. REVERSE is effective only when the execution command is entered.
Input 3	Stop command : STOP STOP stops operation of designated axis.
Output 1	During operation : RUN RUN is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : END OF ZERO POINT RETURN OPERATION When execution is ended without an error, only 1 scan turns ON.

5.17.2.7 Jog Operation (JOG)

The JOG command is used for moving the machine by means of the manual, continuous feed. For feed speed, designate one from parameter settings for speed (1ST) to speed (4TH).



Entering the START command will execute JOG operation while START is on, in a designated derection and with a selected speed.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

4XXXX	Module No.
4XXXX+1	Axis No.
4XXXX+2	Speed No.
4XXXX+3	System use

IF66 No. : 1 to 4

Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)

Feed Speed No. : 1 to 4 (See Table 5.133.)

Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.ssSimultaneous designation of multiple axes is impossible.

4XXXX+1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1

Axis No.

*2 System use (Do not change on ladder.)

4XXXX+3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Execution flag 1 : During execution

CAUTION: Never change set data during execution.

JOG

Data Monitored

Current values and error status are displayed.

4YYYY	Current value — H
4YYYY+1	Current value — L
4YYYY+2	Status

0 to ± 9 9 9 9 9 9 9 9 (DEC)
4YYYY 4YYYY+1

It is displayed if an error occurs.

Refer to Par. 5.17.2.16.

When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is given with a normal open contact (⊣ ⊢). JOG operation is executed while Input 1 is ON and stopped with its OFF.
Input 2	Not used : NONE
Input 3	Reverse rotation command : REVERSE REVERSE designates the direction of rotation : OFF for forward running, and ON for reverse running. REVERSE is effective only when the execution command is entered.
Output 1	During operation : RUN RUN is ON during the execution of a command. It turns OFF when an operation ends and STOP process ends.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : END OF JOG OPERATION When execution is ended without an error, only 1 scan turns ON.

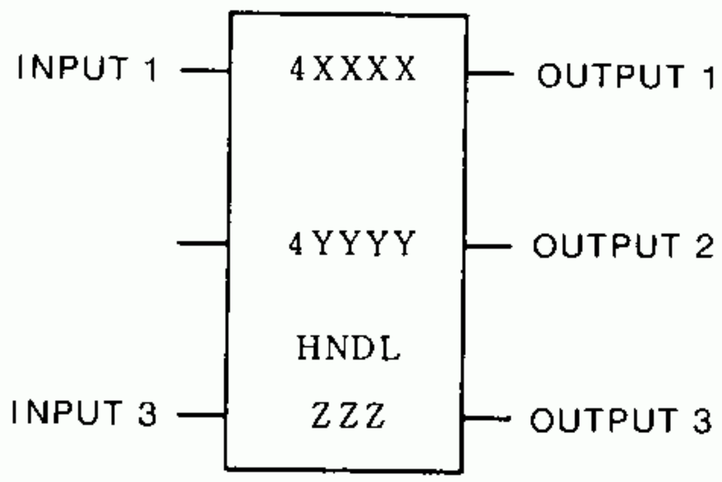
Set the following parameters.

Table 5.133 Speed No. and Feed Speed

	Parameter No.	Parameter Name	Setting Range	Unit
Speed No. 1	P04	Feed Speed (1st)	0 to 240000	× 1000 Command unit/min.
Speed No. 2	P31	Feed Speed (2nd)	0 to 240000	
Speed No. 3	P32	Feed Speed (3rd)	0 to 240000	
Speed No. 4	P33	Feed Speed (4th)	0 to 240000	

5.17.2.8 Handle Operation (HNDL)

When executing handle operation, use the HNDL command to select the handle operation mode. Set parameters of Servopack so as to obtain (P19 - b3 = 1) i.e. use of the pulse reference input. Turning the handle moves the machine in the positive or negative direction according to turning directions. Travel distance per handle scale can be selected from 3 steps of ×1, ×10, and ×100 through switching external inputs (SP 2ND, SP 3RD) of Servopack.



Entering the execution command will switch the mode of designated axes to the handle operation mode, which enables handle operation.

To return the mode to the previous one, enter the cancel command (Turn Input 3). When cancel command is entered, handle operation ends.

Note Set ZZZ to any value from 1 to 999 (DEC).

Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit 7 bit.
Simultaneous designation of multiple axes is impossible.

4XXXX+1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1

Axis No.

*2 System use (Do not change on the ladder.)

4XXXX+2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Execution flag 1 : During execution

CAUTION: Never change set data during execution.

HNDL

Data Monitored

Current values and error status are displayed.

4YYYY	Current value — H
4YYYY+1	Current value — L
4YYYY+2	Status

0 to ± 9 9 9 9 9 9 9 9 (DEC)
4YYYY 4YYYY+1

It is displayed if an error occurs.

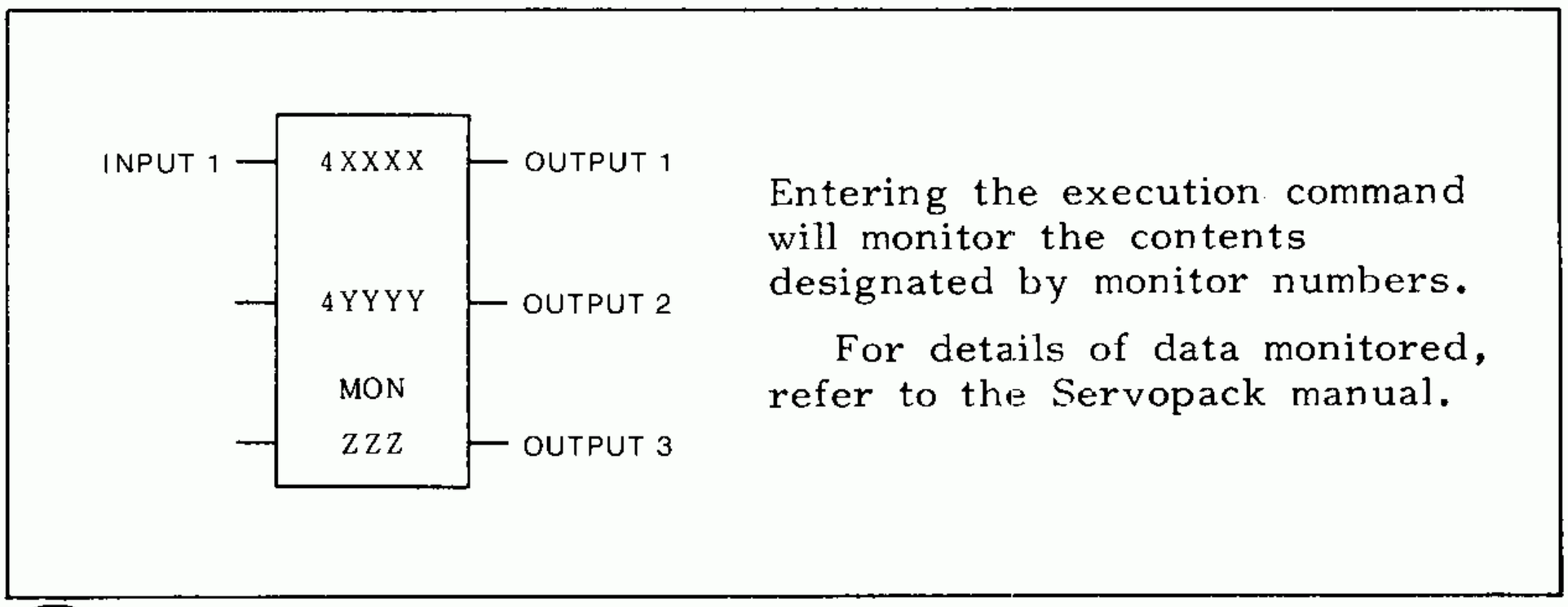
Refer to Par. 5.17.2.16.

When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable. Once execution is started, the execution (monitoring of current position and status) continues until canceled.
Input 2	Not used : NONE
Input 3	Cancel command : MODE CANCEL This command cancels the handle command.
Output 1	During operation : RUN RUN is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error (mode cancellation is ended), only 1 scan turns ON.

5.17.2.9 Monitor (MON)

Executing the monitor command enables the monitor of designated axes.

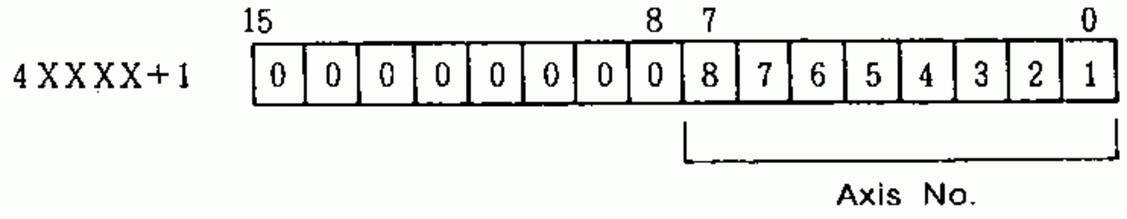


Note Set ZZZ to any value from 1 to 999 (DEC).

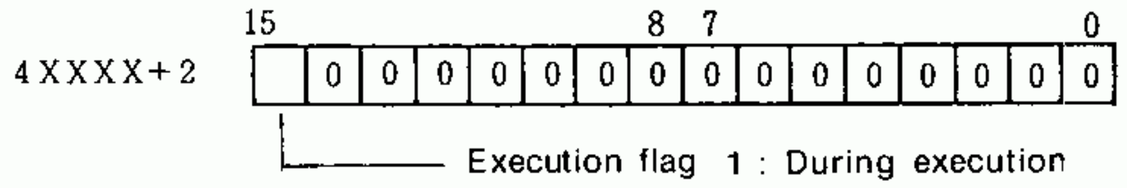
Data to be set

4XXXX	Module No.	1F66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	Monitor No.	Monitor No. : See Table 5.134.
4XXXX+3	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

MON

Monitor Number

Monitor number designates contents to be monitored.

Table 5.134 Monitor No. and Contents

Monitor No. (DEC)	X	Contents
000X	1	Current position
	2	Position deviation
	3	Current speed
	4	Reference speed
	5	Torque reference
	6	Status
001X	0 to 9	0 : Current alarm 1 to 9 : Past alarms before 1 to 9 times
002X	1 to 6	Status of IN 1 to IN 6
003X	1 to 3	Status of OUT 1 to OUT 3
01XX	1 to 99	See servo parameter list
0200		Program No. being currently executed
0300		Variables H1, H2, H3, H4 data

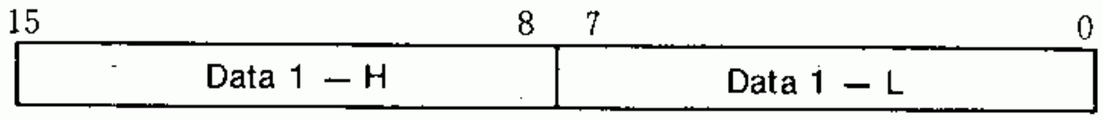
Data Monitored

Current values and error status are displayed.

4YYYY	Data 1 — H
4YYYY+1	Data 1 — L
4YYYY+2	Data 2 — H
4YYYY+3	Data 2 — L
4YYYY+4	Data 3 — H
4YYYY+5	Data 3 — L
4YYYY+6	Data 4 — H
4YYYY+7	Data 4 — L
4YYYY+8	Status

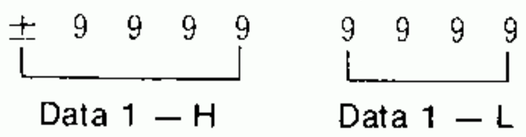
—— It is displayed if an error occurs.
Refer to Par. 5.17.2.16.

For monitor data corresponding to the monitor No., 2 words of Data 1-H and Data 1-L are used (except for Monitor No.300). In case of [-] , 1 is displayed in MSB of Data 1-H.



Example

- Numeric data of current position and beyond are displayed as shown below.



- Bit data of I/O and parameters are displayed as shown below.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During execution : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

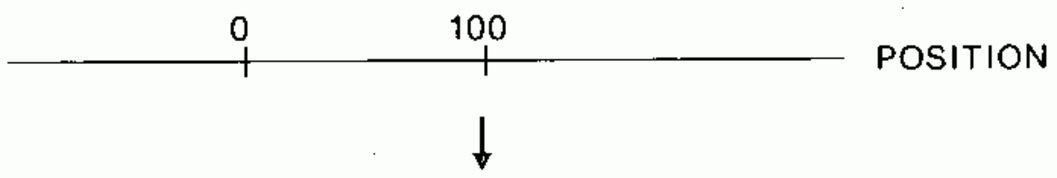
POS

5.17.2.10 Current Position Setting (POS)

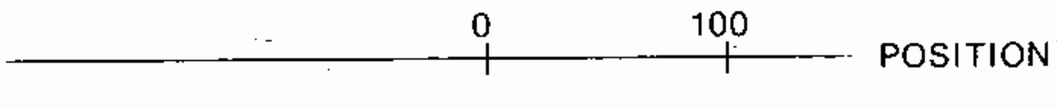
POS is the command for rewriting current positions. After the execution of POS, the current coordinate is changed to a new coordinate.

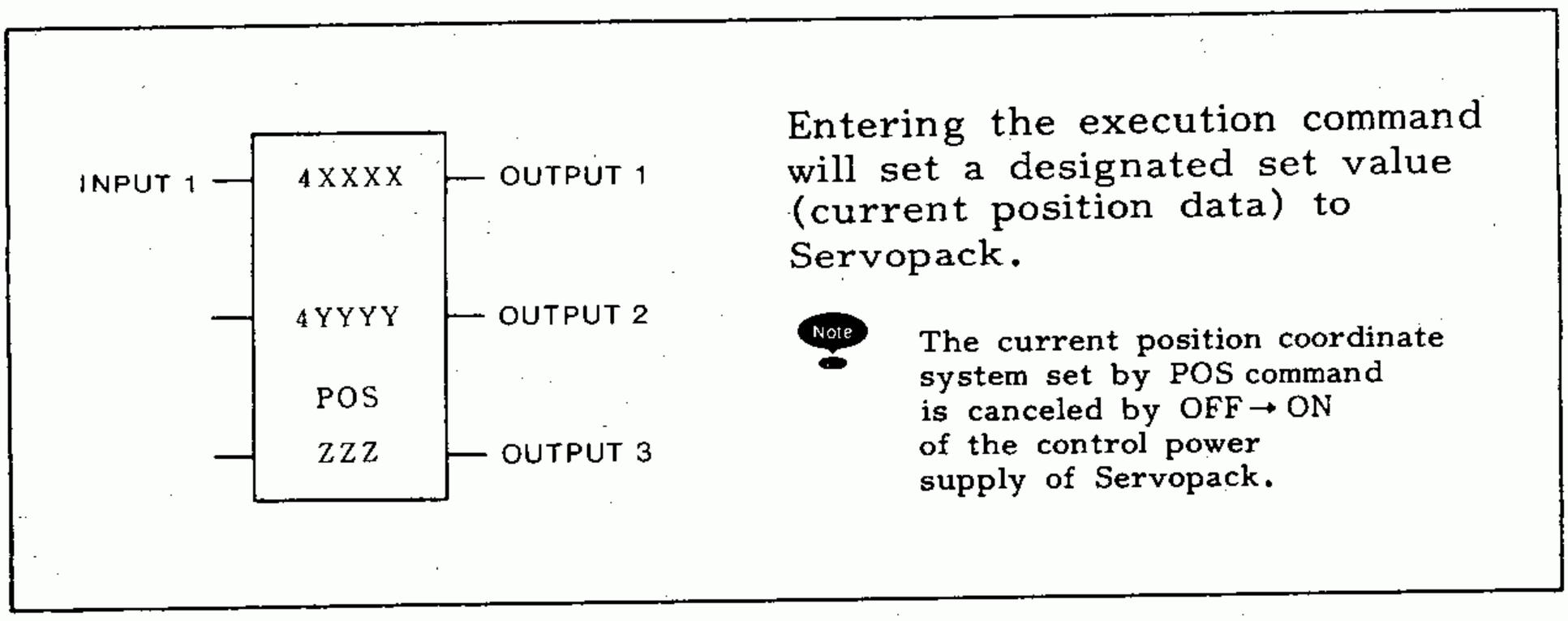
Example

A coordinate with a fixed machine zero point.



Current position is changed to zero.





Data to be set

4XXXX	Module No.
4XXXX+1	Axis No.
4XXXX+2	Set value — H
4XXXX+3	Set value — L
4XXXX+4	System use

IF66 No. : 1 to 4

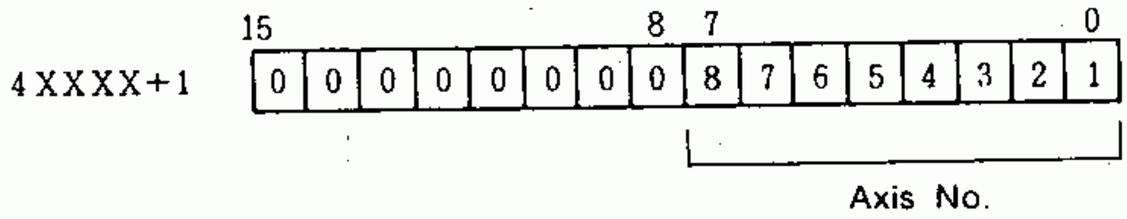
Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)

0 to ± 9 9 9 9 9 9 9 9 (DEC) When MSB of Set value-H is 1, the value will be negative.

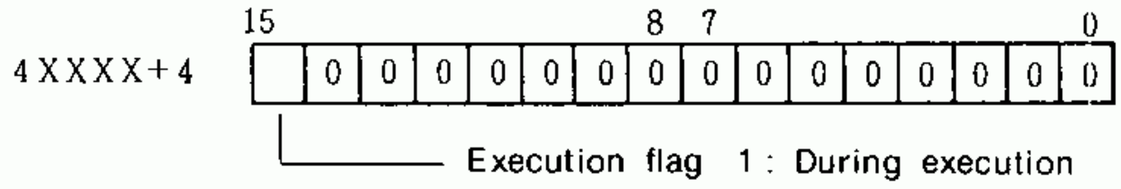
Set value-H Set value-L

Used as execution flag in systems (*2)

Note *1 **Set 1 to an axis to be designated from 0 bit to 7 bit. Simultaneous designation of multiple axes is impossible.**



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

Data Monitored

Error status is displayed.

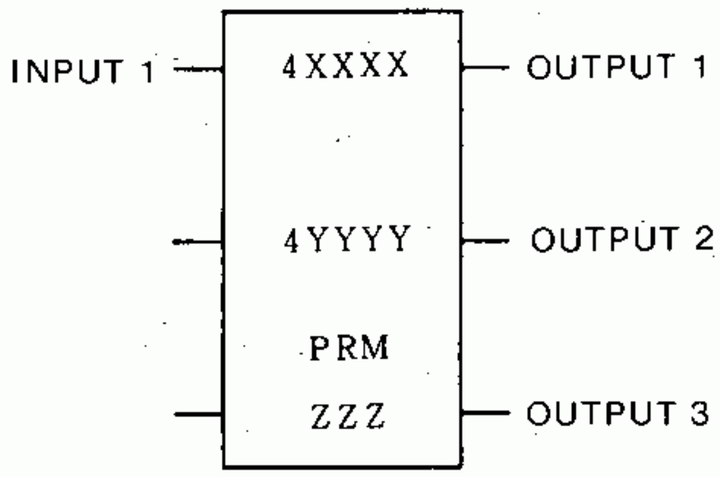
4YYYY Status It is displayed if an error occurs.
Refer to Par. 5 17.2.16.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During operation : RUN RUN is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

5.17.2.11 Parameter Setting (PRM)

PRM is the command for rewriting parameters of Servopack having parameter No. PXX.

Parameters can be rewritten during operation. The parameter No. that can be rewritten by the PRM command are limited.



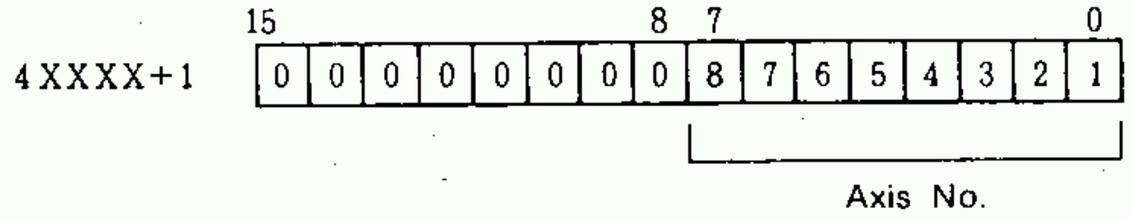
Entering the execution command will set designated parameters to Servopack.

Note Care should be taken that the contents set by the PRM command are changed to the original ones by OFF → ON of the control power supply of Servopack. To keep the contents set by the PRM command, they should be loaded in Servopack.

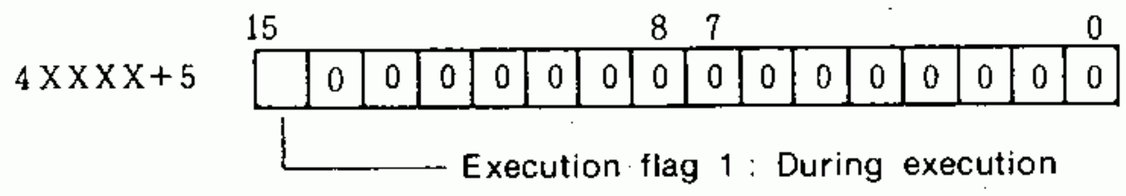
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	Parameter No.	Parameter No. of Servopack : 1 to 100 (DEC)
4XXXX+3	Set value — H	0 to ± 9 9 9 9 9 9 9 9 (DEC) (Set value-H / Set value-L) When MSB of Set value-H is 1, the value will be negative.
4XXXX+4	Set value — L	
4XXXX+5	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

Data Monitored

Error status is displayed.

4YYYY **Status** **It is displayed if an error occurs. Refer to Par. 5.17.2.16.**

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During operation : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

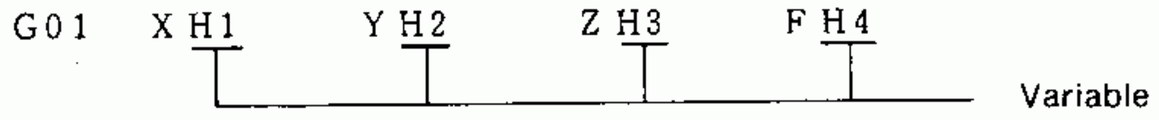
Note **If a parameter data beyond the setting range is set, Servopack takes the maximum value of the parameter effective data range as the set value, without performing error processing.**

VAR

5.17.2.12 Variable Setting (VAR)

Motion programs can be created by using variables H1, H2, H3, and H4. VAR sets the values of these variables.

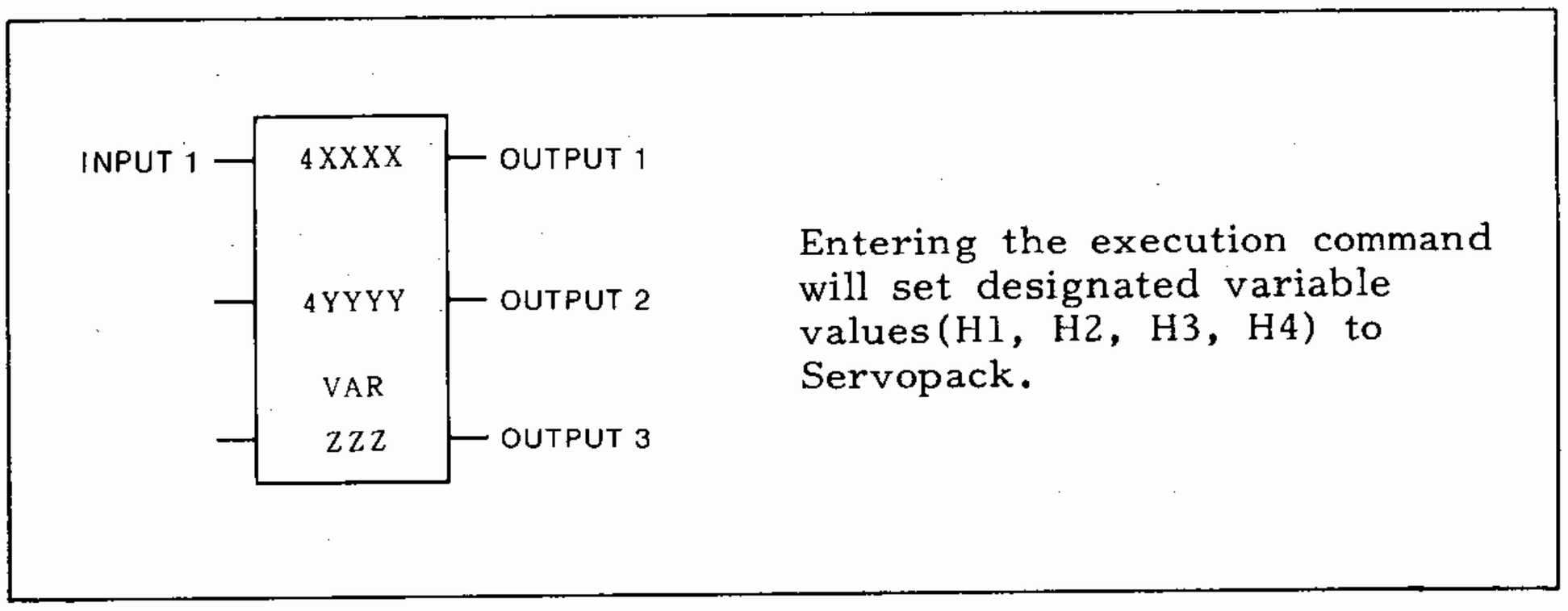
Example



Variable values once set are maintained in Servopack (Battery backup).

However, the use as shown below is impossible.

Example



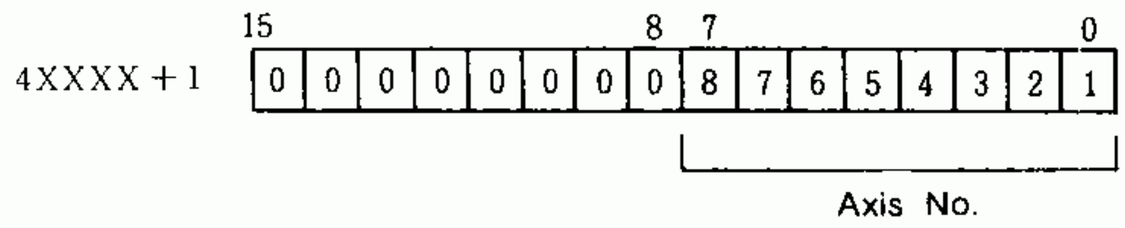
Note The VAR command cannot start with any other command given simultaneously to the same axis, however, the following method enables the start of VAR during the execution of designated axes.

In setting Servopack axis No., set 1 to MSB.

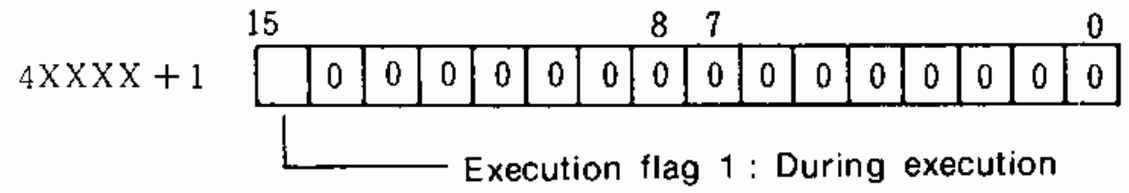
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	H1 Data — H	0 to ± 9999 9999 (DEC) When MSB of Data-H is 1, the value will be negative.
4XXXX+3	H1 Data — L	[Data-H] [Data-L]
4XXXX+4	H2 Data — H	
4XXXX+5	H2 Data — L	
4XXXX+6	H3 Data — H	
4XXXX+7	H3 Data — L	
4XXXX+8	H4 Data — H	
4XXXX+9	H4 Data — L	
4XXXX+10	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.



*2 System use (Do not change on the ladder.)



Note
1. Never change set data during execution.
2. Simultaneous designation of multiple axes is also possible. Since 4YYYY error status reflects only one axis of Servopack having the smallest No., monitoring all the designated axes is impossible.

VAR

Data Monitored

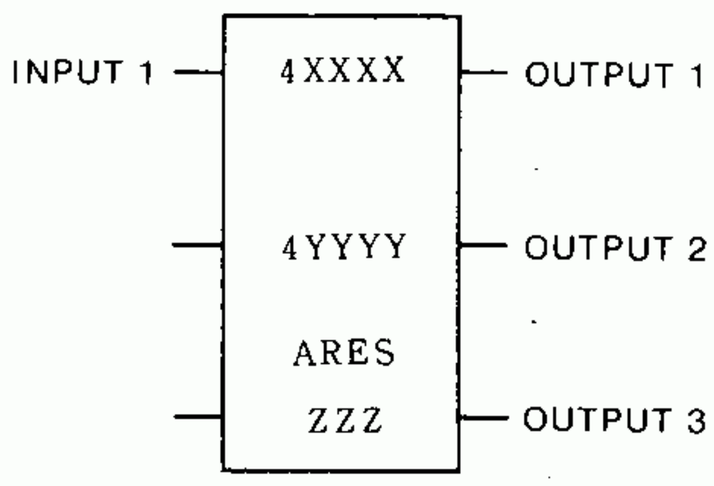
Error status is displayed.

4YYYY Status It is displayed if an error occurs.
Reter to Par. 5 17.2.16

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During operation : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

5.17.2.13 Alarm Reset (ARES)

When Servopack issues an alarm, the ARES command resets the alarm. (The same action as that of Servopack alarm reset signal RST.)



Entering the execution command will clear the alarm of a designated axis.

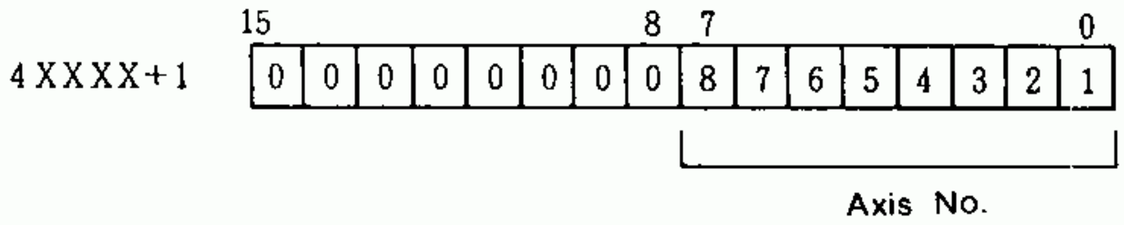
Once an alarm occurs, it is not cleared until it is reset.

Note Set ZZZ to any value from 1 to 999 (DEC).

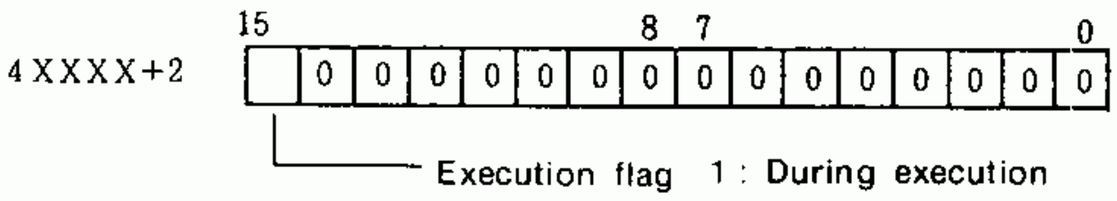
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

ARES

Data Monitored

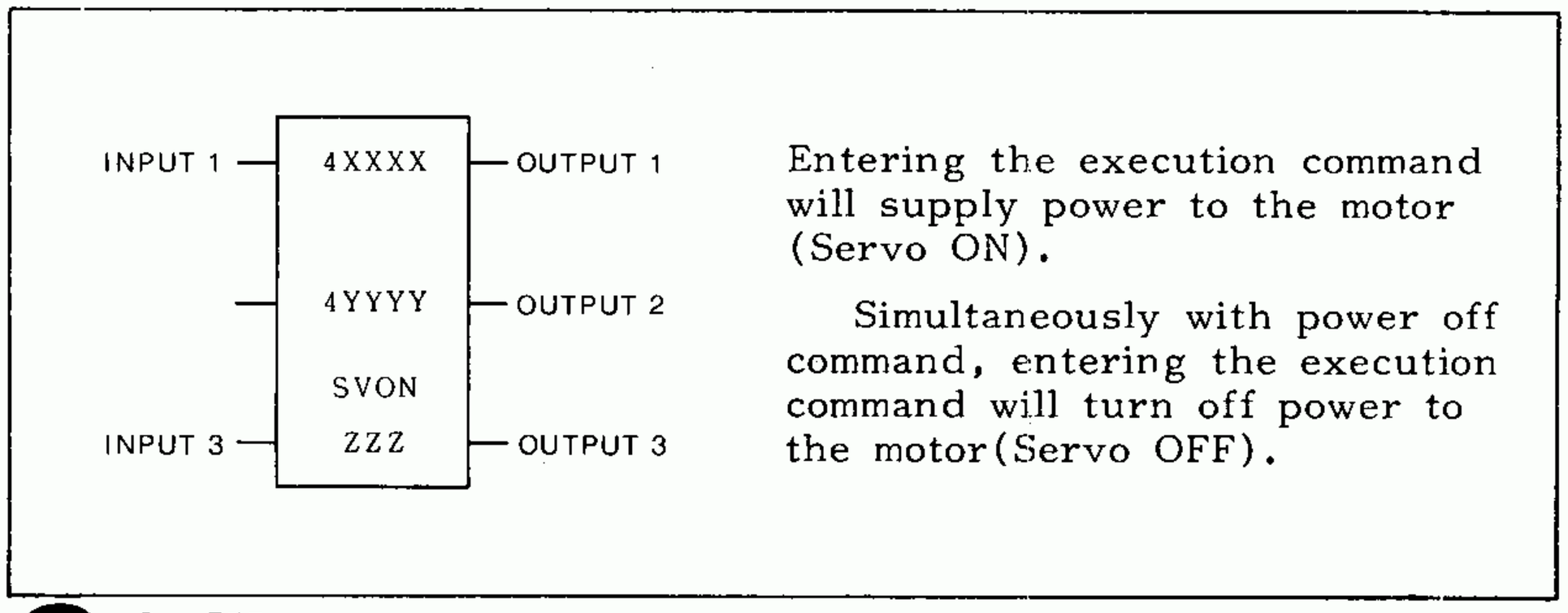
Error status is displayed.

4YYYY Status It is displayed if an error occurs.
Refer to Par. 5.17.2.16.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During operation : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

5.17.2.14 Servo ON (SVON)

The SVON command actuates the power drive circuits of Servopack to supply power to the motor.



Entering the execution command will supply power to the motor (Servo ON).

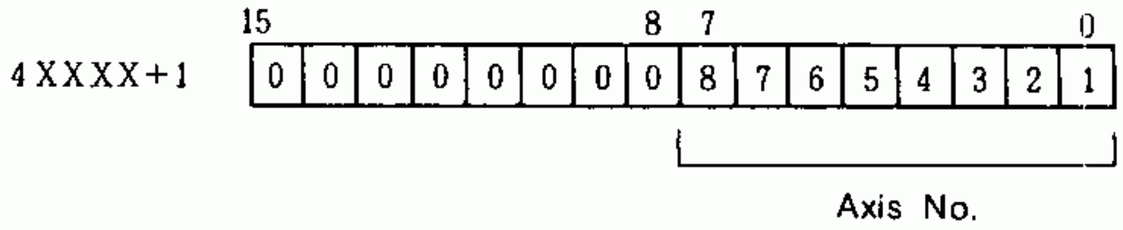
Simultaneously with power off command, entering the execution command will turn off power to the motor(Servo OFF).

Note Set ZZZ to any value from 1 to 999 (DEC).

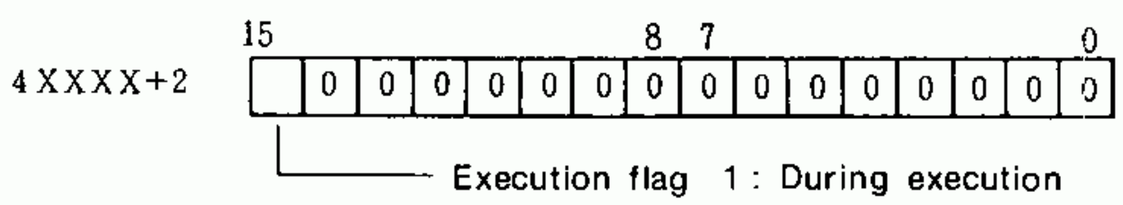
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to a bit of the low-order byte (*1)
4XXXX+2	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to an axis to be designated from 0 bit to 7 bit.
Simultaneous designation of multiple axes is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

SVON

Data Monitored

Current values and error status are displayed.

4YYYY	Current value — H
4YYYY+1	Current value — L
4YYYY+2	Status

0 to ± 9 9 9 9 9 9 9 9 (DEC)
4YYYY 4YYYY+1

It is displayed if an error occurs.

Refer to Par. 5.17.2.16.

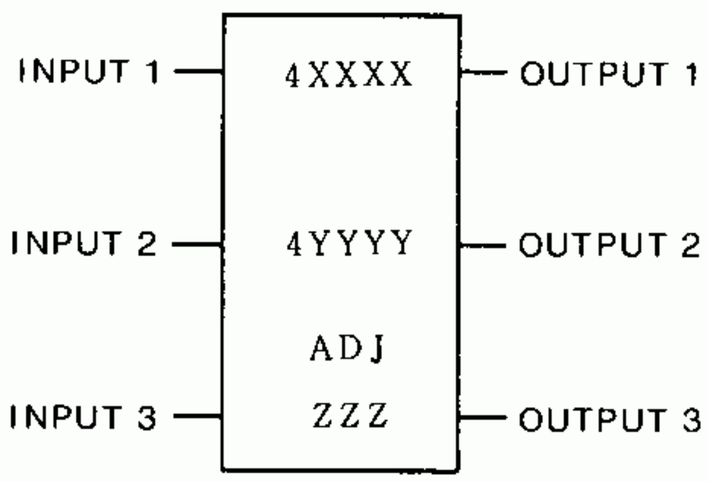
When MSB of 4YYYY is 1, the value will be negative.

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Power off command : OFF OFF is commanded when Input 1 is ON with Input 3 is ON.
Output 1	During operation : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

5.17.2.15 2– or 3– Axis Current Position Adjustment (ADJ)

When executing interpolating operations by using 2 or 3 axes, the ADJ command will adjust current positions among 2 or 3 axes. Because each Servopack controls only its own current position during operation, X-, Y-, and Z-axis must be taught the current positions of 3 axes for interpolating motion programs: G01, XX1, YY1, ZZ1, and FF1.

The purpose of this is to enable Servopack to compute the feed speed of its own axis(axes) from the travel distances of X-, Y-, and Z-axes.



Entering the execution command will monitor the current positions of designated 2 or 3 axes and teach each axis the current positions of 2 or 3 axes.

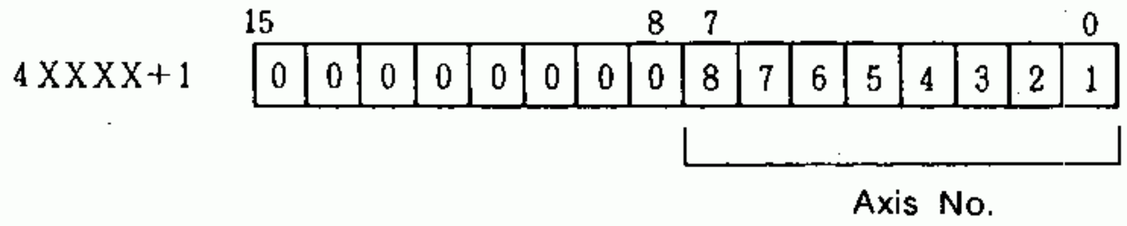
Note Execute this command before executing MOVL and MVLS.

Note Set ZZZ to any value from 1 to 999 (DEC).

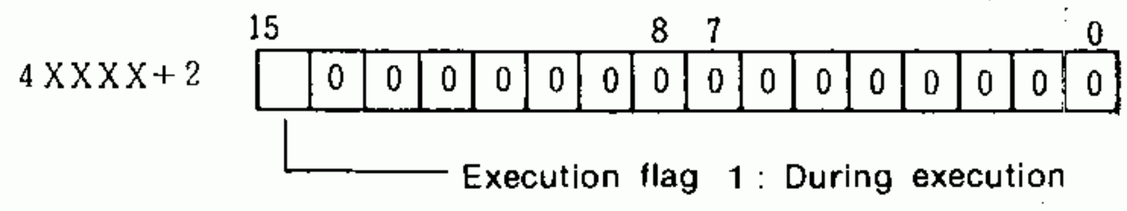
Data to be set

4XXXX	Module No.	IF66 No. : 1 to 4
4XXXX+1	Axis No.	Axis No. of Servopack : Set 1 to 2 or 3 bits of the low-order byte (*1)
4XXXX+2	System use	Used as execution flag in systems (*2)

Note *1 Set 1 to 2 or 3 axes to be designated from 0 bit to 7 bit. Designation of an axis is impossible.



*2 System use (Do not change on the ladder.)



CAUTION: Never change set data during execution.

Data Monitored

Current values and error status are displayed from the smallest number of designated axes.

4YYYY	Current value 1 — H	0 to ± 9 9 9 9 9 9 9 9 (DEC) 4YYYY 4YYYY+1
4YYYY+1	Current value 1 — L	When MSB of 4YYYY is 1, the value will be negative.
4YYYY+2	Current value 2 — H	
4YYYY+3	Current value 2 — L	
4YYYY+4	Current value 3 — H	
4YYYY+5	Current value 3 — L	
4YYYY+6	Status 1	It is displayed if an error occurs. Refer to Par. 5.17.2.16.
4YYYY+7	Status 2	
4YYYY+8	Status 3	

Input 1	Execution command : START START is commanded with rise differentiation (⊣↑⊢). Its ON/OFF during execution is unacceptable.
Input 2	Not used : NONE
Input 3	Not used : NONE
Output 1	During operation : BUSY BUSY is ON during the execution of a command. It turns OFF when the mode cancel is done.
Output 2	Error : ERROR If an error occurs, only 1 scan turns ON.
Output 3	End : DONE When execution is ended without an error, only 1 scan turns ON.

5.17.2.16 Status

For the status of holding registers, the status after start is displayed.

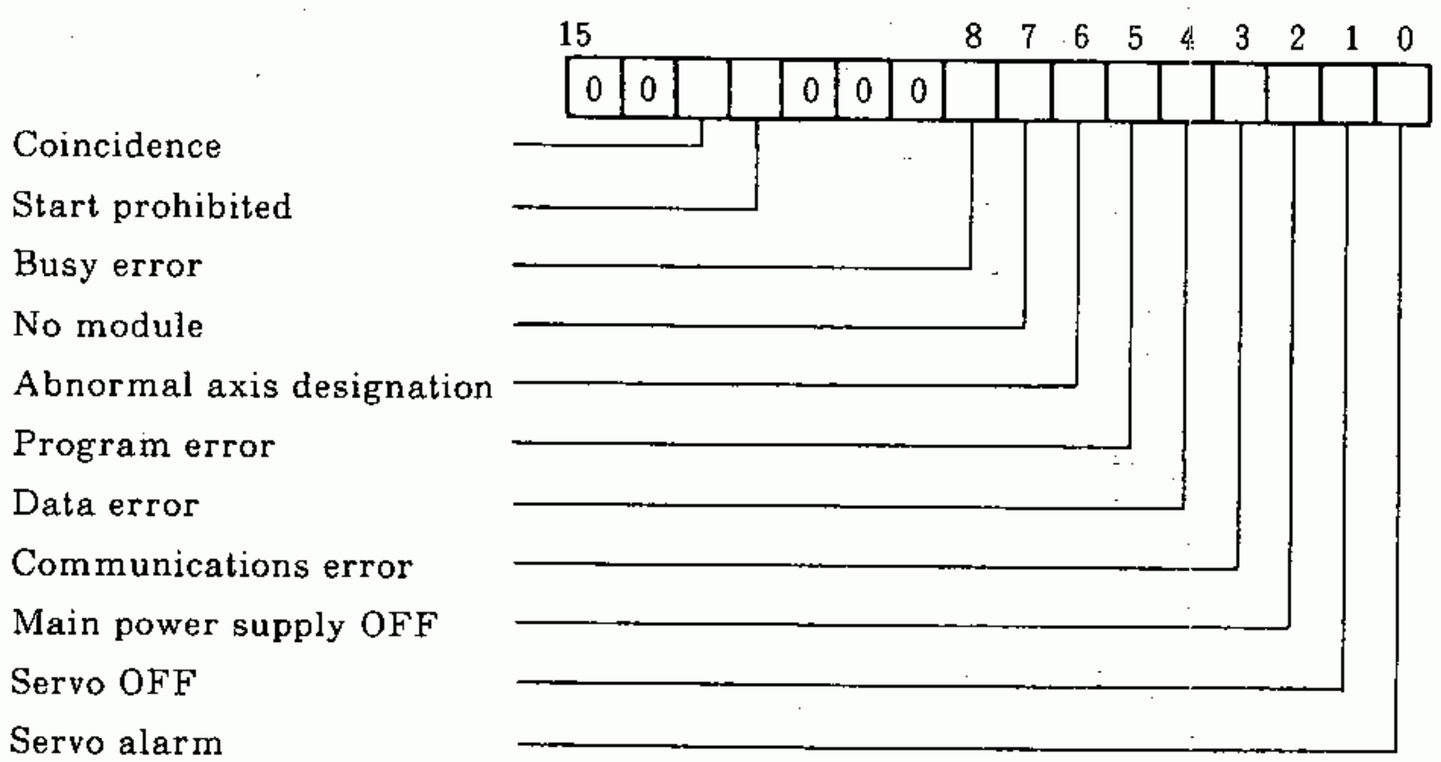


Table 5.135 Status

	Item	Bit Data	Contents
b0	Servo Alarm	1	Alarm occurs at the servo side. Alarm content is monitored by the MON command.
b1	Servo OFF	1	Power is not supplied to the motor. Servo off status
b2	Main Power Supply OFF	1	Servo power supply is not provided. Time out
b3	Communications Error	1	Error occurs in communication between IF66 and servo.
b4	Data Error	1	Error occurs if Error occurs in set data, Module No. is out of range (other than 1 to 4), O# is out of range (other than 1 to 20), N# is out of range (other than 1 to 999), or designated O# or N# does not exist.
b5	Program Error	1	Error occurs in motion programs.
b6	Abnormal Axis Designation	1	Axis No. out of range. Axis designation of 2 or 3 axes does not match with program.
b7	No Module	1	Designated IF66 module does not exist.
b8	Busy Error	1	Other motion command is in execution.
b12	Start prohibited	1	This is indicated during loading or saving of motion programs and parameter.
b13	Coincidence	1	Distance difference (in command unit) between the current position and the target value given by command data is within the range set by Parameter 6 (P06).

SECTION 6
I/O ALLOCATION

I/O allocation is to define the relation between I/O reference numbers used in programming (software) and I/O modules mounted on the control panel (hardware).

Since I/O modules can be located at any module slot corresponding to I/O allocation, a variety of combination of I/O modules is available. Before operating GL40S Controller, be sure to set the I/O allocation table to the CPU module memory using the P150 or P140 programming panel. I/O allocation is made under the condition that CPU module is stopped (Scan Stopping).

I/O allocation is made independently to each location. A change of I/O allocation made to a location does not affect those for the other locations. For the operation of I/O allocation, refer to the "P150, P140 Programming Panel User's Manual" (SIE-C815-15.2, -15.3).

Note For COMM module, PC link module and Servo interface module, I/O allocation is unneccesary.

6.1 I/O CONFIGURATION

Fig.6.1 shows the I/O section of GL40S. The maximum slot number is 35.

A total of 512 discrete I/O points and a total of 128 sets of register I/O can be mounted in any location of the I/O modules.

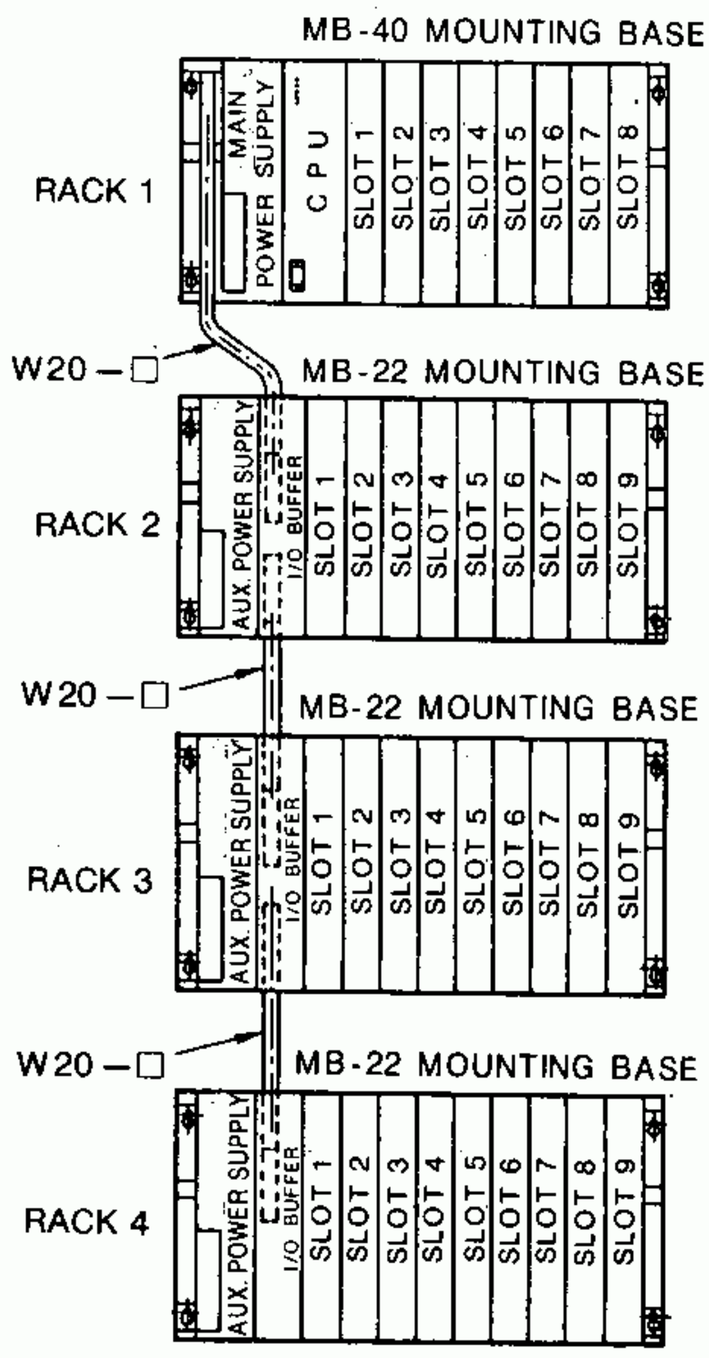
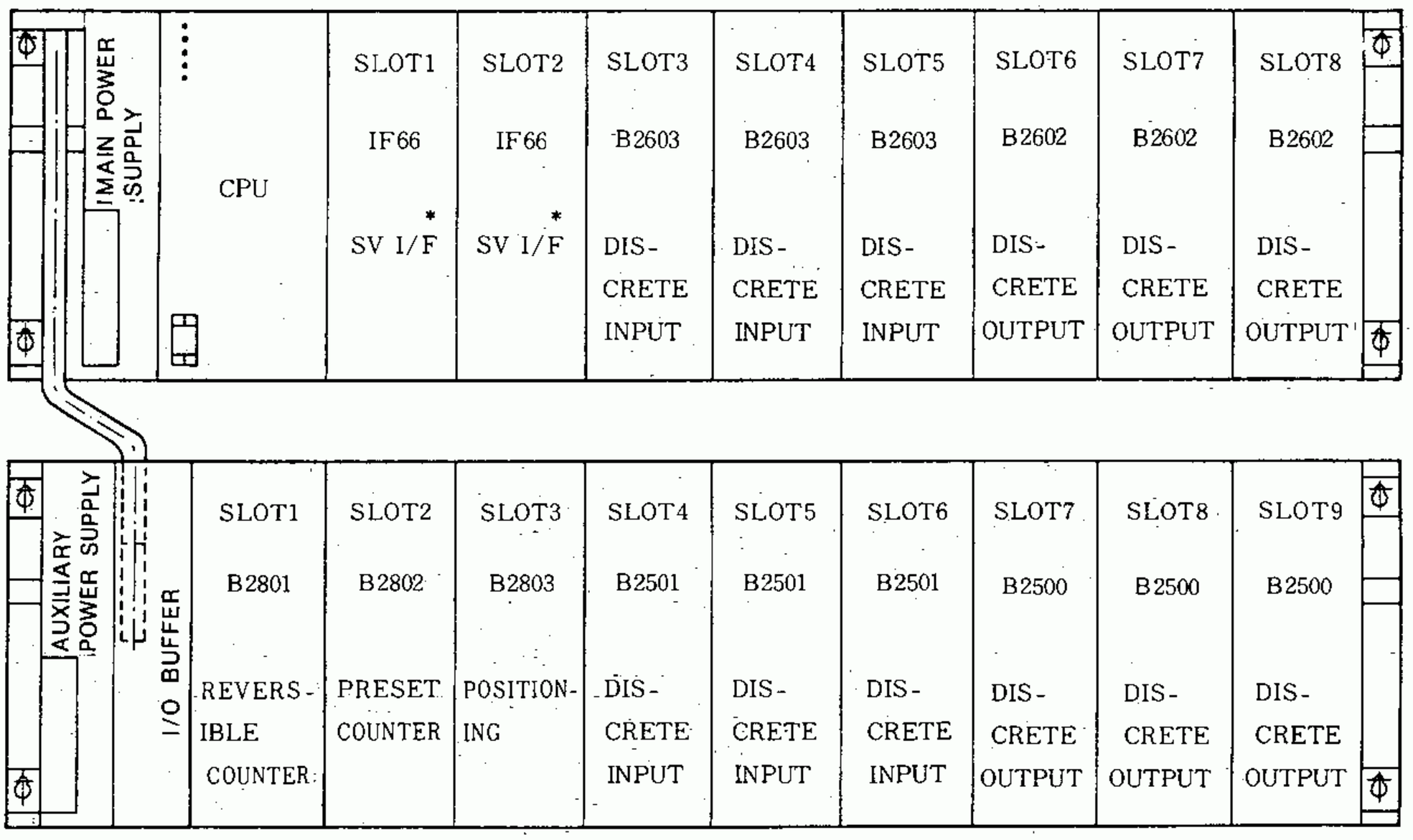


Fig. 6.1 I/O Section Configuration

6.2 I/O MODULE LAYOUT

I/O modules may be installed at any location of racks 1,2,3 or 4 only in a range of 512 discrete I/O points and 128 register I/O points. They need not be installed in contiguous locations. However, it is recommended that the I/O modules be installed in groups (by input and output, voltage level, application, etc.). Fig. 6.2 shows a sample layout of I/O modules.



* Allocatior is unnecessary.

Fig. 6.2 Sample Layout of I/O Modules

6.3 I/O NUMBERS

I/O signals include discrete inputs/outputs and register inputs/outputs (numerical value). The reference number is used as the I/O number.

Table 6.1 I/O Number List

Input/Output Type	I/O Number (Reference Number)
Discrete Input (Input Relay)	10001-10512
Register Input (Input Register)	30001-30128
Discrete Output (Output Coil)	00001-00512
Register Output (Output Register)	40001-40128

NOTE The table above shows the range of I/O numbers as assigned to each group of I/O signals. Note the following limitations :

- Discrete inputs + discrete outputs ≦ 512
- Register inputs + register outputs ≦ 128

6.4 I/O MODULE LOCATION

The location of I/O module is defined by a rack number and a slot number. Number of slots differs with Rack No.1 and Rack No.2 or more as shown in Table 6.2.

Table 6.2 Location of I/O Module

Rack No.	Slot No.								
1	1	2	3	4	5	6	7	8	–
2	1	2	3	4	5	6	7	8	9
3	1	2	3	4	5	6	7	8	9
4	1	2	3	4	5	6	7	8	9

It is permissable to assign a pair of discrete I/O or a pair of register I/O to a slot. This is possible owing to the modules, such as the counter, PID, and positioning module, each of which deals with discrete I/O and register I/O, (called a modular module). The number of I/O points allocated to a slot is given in Table 6.3.

Table 6.3 Number of I/O Points Allocated to a Slot

I/O Type	Allowable I/O Points in Allocation
Discrete Input	8, 16, 24, 32, 64
Discrete Output	8, 16, 24, 32, 64
Register Input	1 – 8
Register Output	1 – 8

Note

1. Any I/O allocation is available in the range of I/O points given above. To a slot where a 16-point discrete output module is installed, for example, 8 or 16 discrete outputs must be allocated. If 24 or 32 points are allocated to the slot (it is possible through the P190 programming panel), the outputs may be degraded.
2. External data are stored as is in input registers as register inputs. If input as BCD, internal processing does not perform correct arithmetic operations unless binary conversion is conducted using the arithmetic function BIN. Similarly, register output must also be output by BCD after converting it to BCD using the arithmetic function BCD.

6.5 TYPES OF I/O MODULES AND I/O ALLOCATION

The maximum number of I/O points for allocation is specified in accordance with the type of I/O module as shown in Table 6.4. Refer to Table 6.4 and the following.

- Registers can be allocated, in 16-bit binary form, to a discrete module.
- I/O allocation for the register module can not be performed in discrete form. Register module I/O data are given in 16-bit binary form.
- Analog modules can not be allocated in discrete form.
- Both discrete and register points must be allocated to compound modules such as counters and positioning modules.
- For instrumentation modules, perform I/O allocation only in the right slot.

Table 6.4 Number of I/O Points for Allocation by Module Type

Modules	Type	Input Point No.		Output Point No.	
		Discrete	Register	Discrete	Register
16-point Discrete Input	B2501, B2503 B2601, B2611	16	0	0	0
		0	1	0	0
32-point Discrete Input	B2505, B2507 B2603, B2607	32	0	0	0
		0	2	0	0
64-point Discrete Input	B2605	64	0	0	0
		0	4	0	0
Register Input	B2701	0	8	0	0
Analog Input	B27□3	0	8	0	0
16-point Discrete Output	B2500, B2600 B2904, B2914	0	0	16	1
		0	0	16	1
32-point Discrete Output	B2504, B2602 B2606, B2902	0	0	32	0
		0	0	0	2
64-point Discrete Output	B2604	0	0	64	0
		0	0	0	4
Register Output	B2700	0	0	0	8
Analog Output	B27□2	0	0	0	2
Reversible Counter*	B2801	16	2 or 4	8 or 16	2 or 4
Preset Counter*	B2802	16	2	16 or 24	2 to 8
Positioning*	B2803, B2813 B2823	16	4	24	4
PID*	B2800	8	8	16	8
Instrumentation	B2705	0	5	0	0
MEMOLINK master*	B2804	128	8	128	8
MEMOLINK slave*	B2805	128	8	128	8
I/F*	B2806	128	8	128	8

*This module is called a modular module.

6.6 I/O ALLOCATION REFERENCE NO.

The reference number should be the first number allocated to the slot. For discrete I/O allocation, the reference numbers must begin with fixed numbers as follows.

- First number of discrete output = 00001 + 8n (n=0, 1, 2, ..., 63)
- First number of discrete input = 10001 + 8n (n=0, 1, 2, ..., 63)

The following limitations exist in relation to the range of the GL40S reference numbers.

- First number of discrete output + number of points ≦ 513
- First number of discrete input + number of points ≦ 10513
- First number of register input + number of points ≦ 30129
- First number of register output + number of points ≦ 40129

6.7 I/O SIGNAL PROCESSING

6.7.1 Input Signals

Discrete inputs are stored in input relay (10001 to 10512) memories, and register inputs in input register (30001 to 30128) memories.

I/O signal processing is determined to first store discrete inputs into input relay memories and secondly register inputs into input register memories.

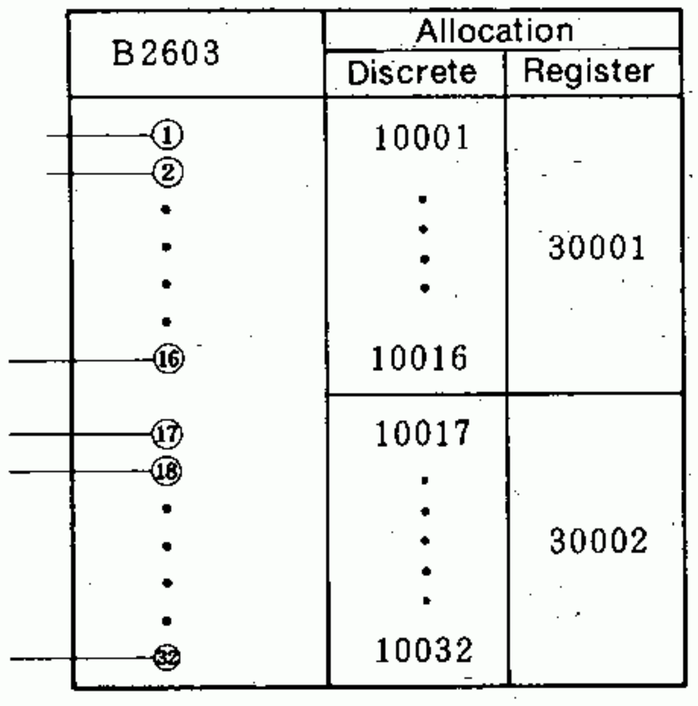
When allocating inputs, it is possible to allocate both discrete and register inputs to the same slot. However, it is impossible to divide the points of a slot into discrete and register inputs.

For example, suppose that 16 ON/OFF signals and one set of numeral data (BCD 4-digit) are input, by the use of a 32-point module (B 2603, B 2505, etc.). Inputs are allocated as follows.

Discrete --- Reference 10001, size 32 points.
Register --- Reference 30001, size 2 sets.

If ON/OFF signals are allocated to the first half and numeral data to the second half, the result is as follows:

Discrete --- 10001 to 10016,
Register --- 30002



In this case, input relays (10017 to 10032) and input register (30001) become idle. Therefore, do not use in the program.

6.7.2 Output Signals

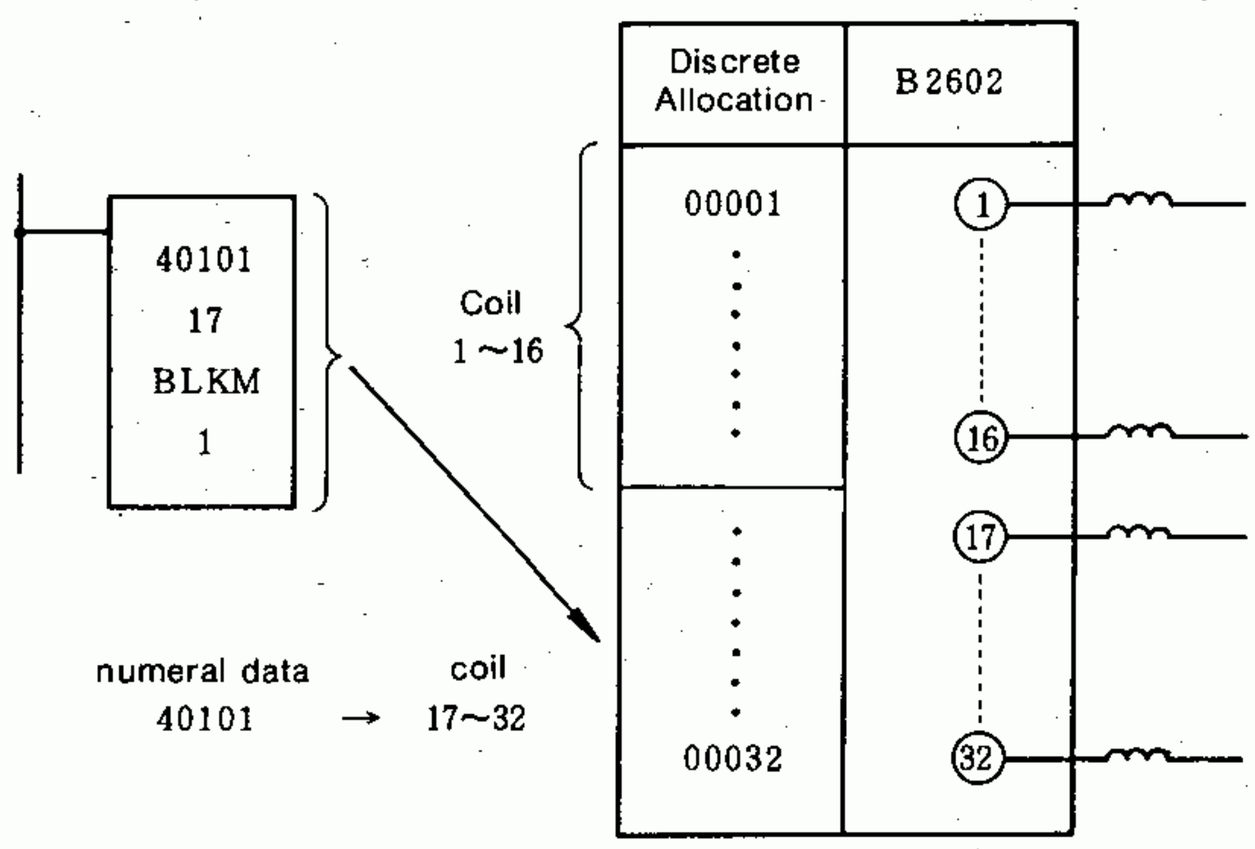
Output signals are stored in output coil (00001 to 00512) memories and output register (40001 to 40128) memories.

Output signal processing first transmits the ON/OFF state of coil memories to the output module and secondly the contents (numeral data) of output register memories to the output module.

If both discrete and register allocations are made to the same slot in allocating outputs, signals from the output module will be disturbed (The contents of coil memories and those of registers will be output alternately.)

It is therefore impossible in principle to allocate both discrete and register signals to the output module of the same slot.

For example, suppose that 16 ON/OFF signals and one set of numeral data (BCD 4-digit) are output by the use of a 32-point module (B2602, B2504, etc.). Output allocation, applied only to discrete signals, is made to the reference 00001, size 32 points. If ON/OFF signals are allocated to the first half and one set of numeral data to the second half, the numeral data are output with coils (00017 to 00032). Therefore, using data transmission (BLKM) in a ladder program, the content (BCD 4-digit) of numeral data is converted to coil memories, before being output.



SECTION 7
ROM OPERATION

ROM operation, that utilizes user programs or holding registers fixed in the ROM, is described. Use the ROM pack (JAMSC-MM40, -MM41).

7.1 WHAT IS ROM OPERATION?

In ROM operation, user programs and holding registers are stored and fixed in a read-only-memory ROM, and the memory contents are transmitted to the RAM in the CPU at the time of power ON to be used for operation. Since the ROM holds its memory contents even when power is OFF, it can protect programs and data. Unless the ROM is changed, programs and data do not change.

The memory type and data transmission range are set by the dip switch in front of the CPU (the battery cover is opened.).

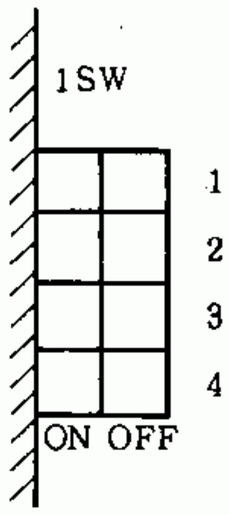


Fig. 7.1 Setting Switch

Table 7.1 Switch Setting and Transmission Data

1SW-1	1SW-2	1SW-3	ROM → RAM Transmission Range			Memory Type	ROM Type
			User Program	I/O Allocation	Holding Register		
OFF	×	×	—	—	—	RAM	—
ON	OFF	OFF	○	○	—	EPROM	MBM27C256 or equivalent
ON	OFF	ON	○	○	○		
ON	ON	OFF	○	○	—	EEPROM	X28C256 or equivalent
ON	ON	ON	○	○	○		

×: Disregarded
○: ROM → RAM Transmission Range

Note: Starting operation from the programming panel does not perform ROM → RAM transmission.

7.1 WHAT IS ROM OPERATION? (Cont'd)

After turning power ON for ROM operation, 1-SW4 specifies whether the CPU immediately enters into operation (move to the RUN mode) or the CPU at a stop waits for the START command from the programming panel.

Table 7.2 CPU Status at the Time of Power ON

1SW-1	1SW-4	Mode	Operation
ON	OFF	RUN	After ROM → RAM transmission, CPU starts.
	ON	STOP	After ROM → RAM transmission, CPU stops. (Starting is commanded from PP.)

Because ROM operation assumes operation without battery, the data status at the time of power ON differs from that of normal RAM operation as shown below.

Table 7.3 Data at the Time of Power ON

1SW-1	1SW-2	1SW-3	Holding Register	Latch Coil	Disable Table	History of coils and input relays
OFF	×	×	Status Held	Status Held	Hold	Renewed
ON	OFF	OFF	Cleared to zero	OFF	Cleared	Cleared
ON	OFF	ON	ROM data	OFF	Cleared	Cleared
ON	ON	OFF	Cleared to zero	OFF	Cleared	Cleared
ON	ON	ON	ROM data	OFF	Cleared	Cleared

History : The state of coils and input relays at the time of the preceding scan.

Note In case of ROM operation, battery check is not performed. Therefore, the battery alarm indicator is always OFF.

7.2 ROM WRITING

A PROM is mounted to the ROM pack beforehand.

Table 7.4 ROM Pack

ROM Pack Type	ROM Type	Access Time	Manufacturer	Remarks
JAMSC-MM40	MBM27C256	250ns	FUJITSU LTD.	IC Socket
JAMSC-MM41	X28C256	250ns	Xicor	Soldered

The writing procedure of program data in the PROM and the procedure of ROM operation are described as follows. For detailed key operation of the programming panel, refer to the "User's Manual of Programming Panel (P150, P140)."

7.2.1 EPROM

The programs of GL40S are transmitted from the programming panel to the PROM writer and are written in an EPROM.
(Data Format: INTEL HEX)

With the PROM writer, the following types (Table 7.5) are recommended. For PROM writer handling, refer to the manual attached to the writer. Use the cable recommended by the maker of the PROM writer.

Table 7.5 Recommended PROM Writer

Manufacturer	PROM Writer Type
ADVANTEST	R4944A
AVAL	PKW-3100

EPROM (MM40) Operation

START

Utilizing the loader function of the programmer, save CPU data in a floppy.

Connect the programming panel with the PROM writer using a special cable.

Utilizing the loader function of the programmer; transmit data in the floppy to the PROM writer.

Remove the EPROM from the ROM pack (JAMSC-MM40).

Set the EPROM to the PROM writer, and write in programs. (See Par. 7.2.2)

Set the EPROM that completed writing to the IC socket of the ROM pack. Be careful of the direction.

Turn OFF power supply to the PC.

Open the battery cover of the CPU to set DIP switch at " EPROM SELECTION ".

1SW-1	1SW-2
ON	OFF

Set 1SW-3, -4 according to purposes.

Take out the empty pack from the CPU.

Mount the ROM pack to the CPU.

Turn ON power supply to the PC.

1SW-4: ON
After ROM → RAM transmission
PC: At a stop(STOP)

1SW-4: OFF
After ROM → RAM transmission
PC: Running state (RUN)

END

* Use EPROM completely erased (over 30min.) by the use of an ultraviolet eraser.
  The EPROM attached to JAMSC-MM40 is delivered after being erased completely.

ROM Operation

7.2.2 PROM Writer Setting

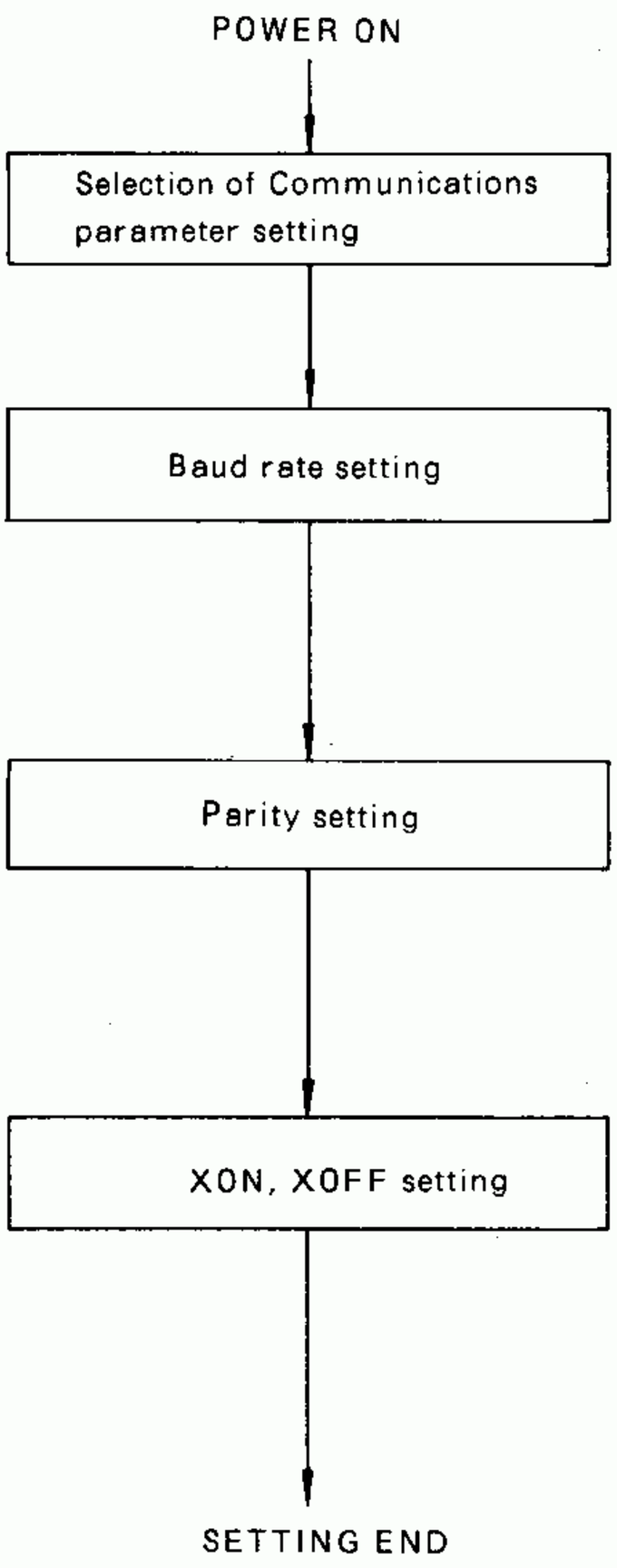
[Example 1]

ADVANTEST R4944A Initial Setting

(1) Transmission Parameter Setting

The transmission parameters on the P150 printer side are made identical with those of the PROM writer. The BAUD switch and the PARITY switch on the side face of the PROM writer are set only at the time of power ON.

Transmission parameters
Baud rate : 9600 bps
Parity : None
Stop bit : 1
Data length : 8 bits



SELECT A

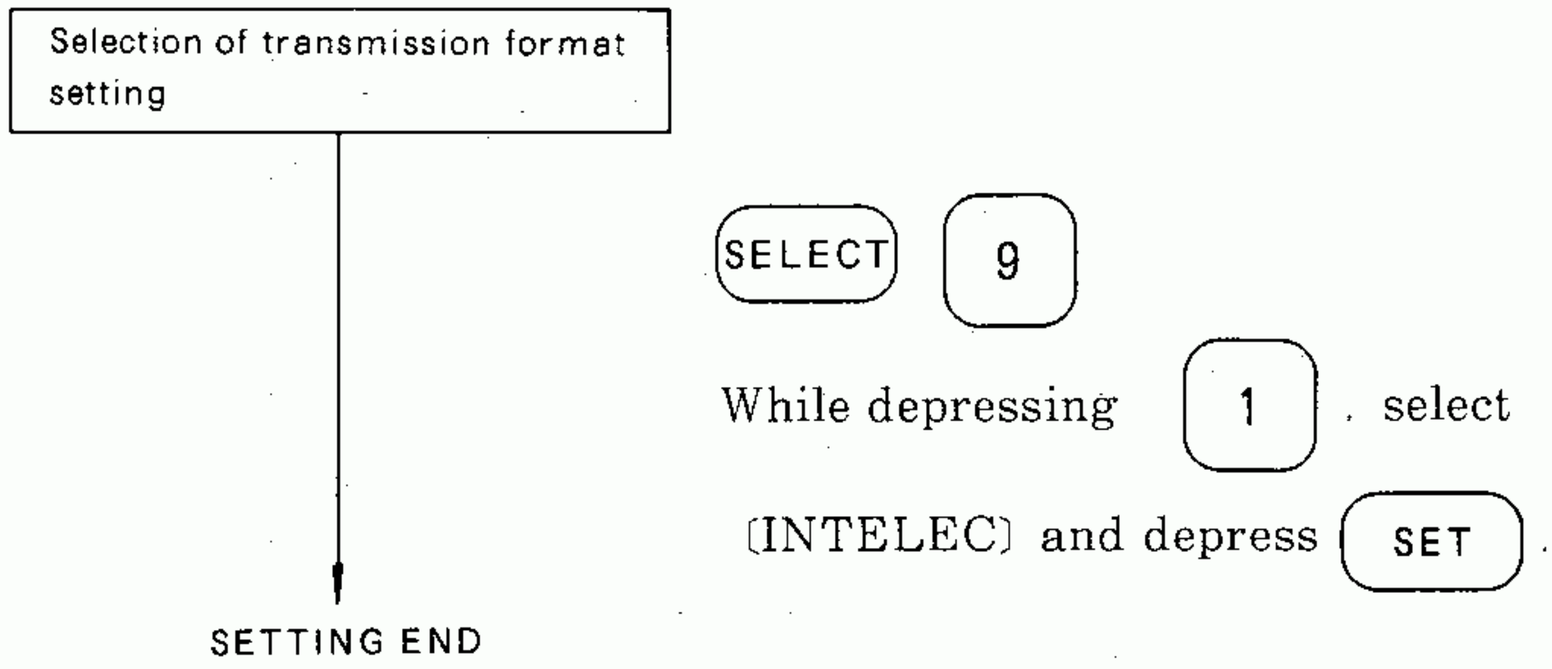
While depressing 1 , select 〔9600〕 and depress △ .

While depressing 1 , select 〔NON〕 and depress △ .

While depressing 1 , select 〔XOF〕 and depress SET

(2) Transmission Format Setting g

INTEL HEX Format is set



(3) Cable Connection (RS232 C connection)

PROM Writer Side

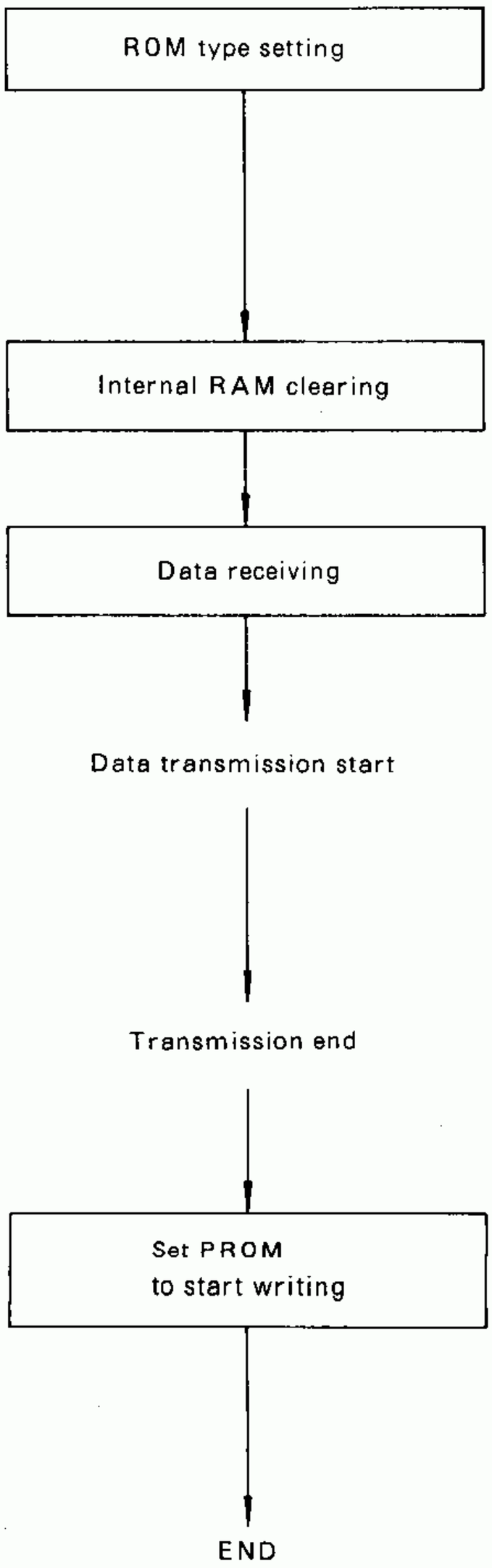
Pin No.	Signals
1	GND
2	TXD
3	RXD
4	RTS
5	CTS
6	
7	SG
20	DTR

(R4944A)

P150 Side

Pin No.	Signals
1	GND
2	TXD
3	RXD
4	RTS
5	CTS
6	DSR
7	SG
20	DTR

ADVANTEST R4944A ROM Writing



ROM TYPE 2 5 6 SET

The example shows the case of HN27256, HN27C256, and MBM27256. The procedure differs among ROM types used.
Refer to the Manual of the PROM writer.

EDIT F SET

EDIT 6 SET

Start transmission from the P150 side in sequence of F→ EPROM → File selection → CONFIRM

DEVICE F SET

DEVICE SET

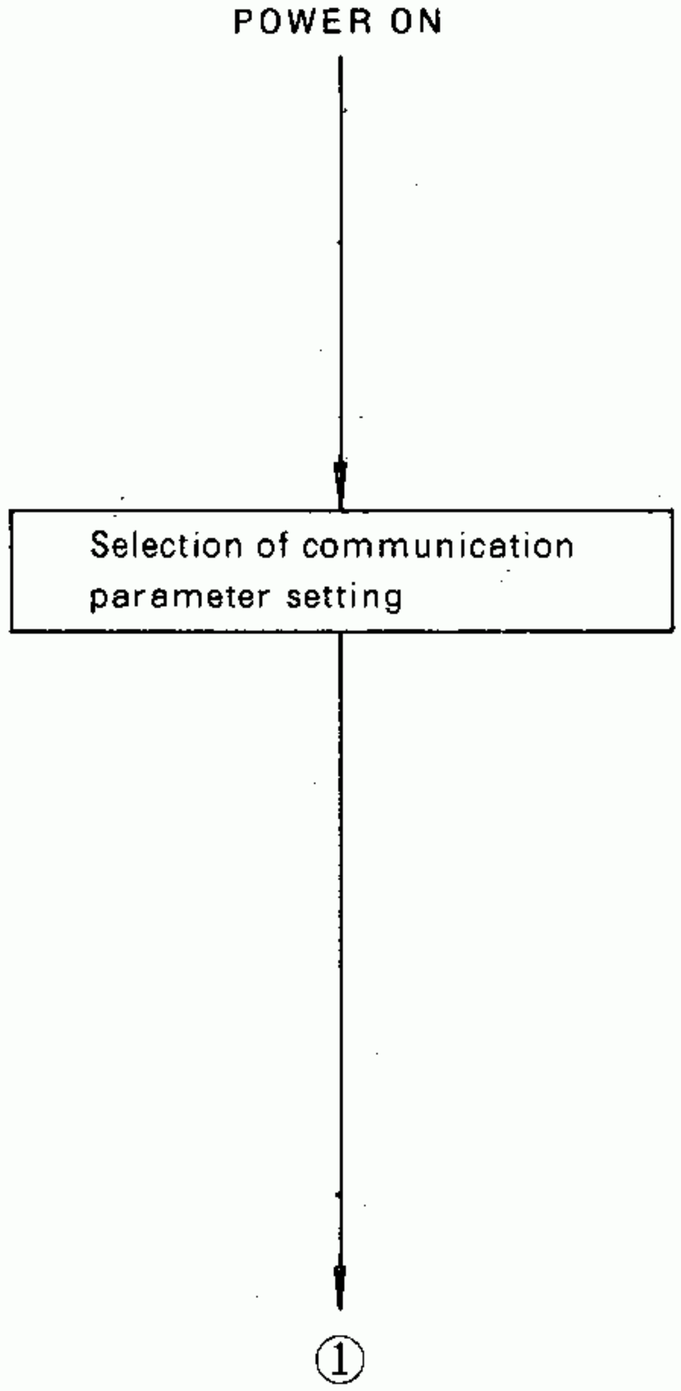
[Example 2]

AVAL PECKER 30 Initial Setting

(1) Transmission Parameter Setting

The transmission parameters on the P150 printer side are made identical with those of the PROM writer.

Transmission Parameters	Baud rate :	9600 bps
	Parity :	None
	Stop bit :	1
	Data length :	8 bits



Display

```
Std Mode
PKW-3100  Ver ×.××
Select        F No.
Prg + Edit  Trn + Set
```

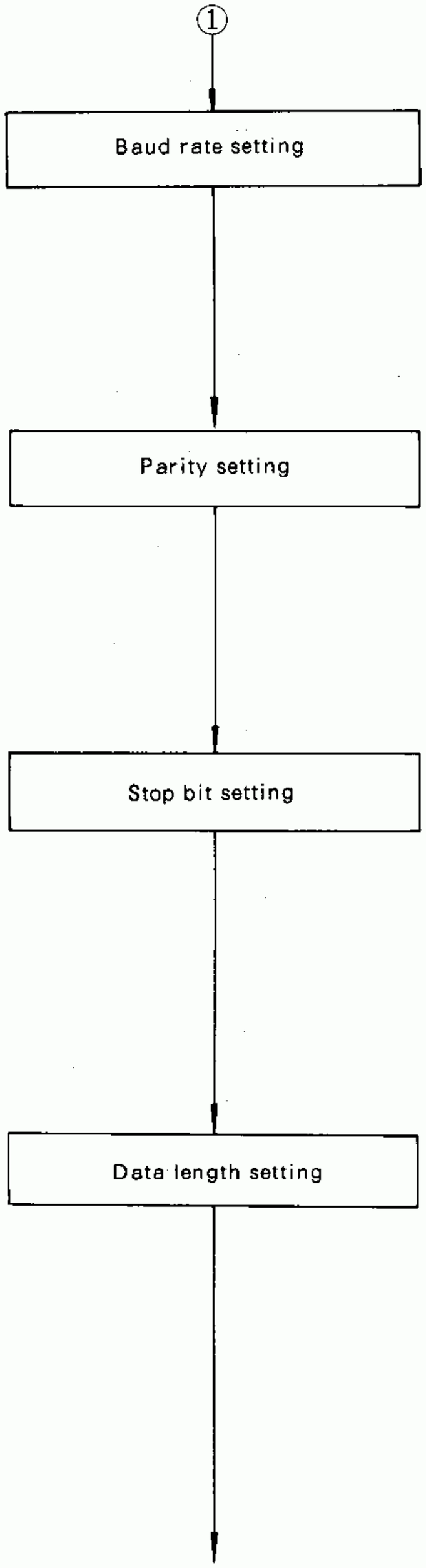
ESC — Initial display (Display on the left)

◁ — Preceding display

Depress F4 (Set) and F2 (COMU).

Display

```
Comunicat
INTEL     27512
F No. or  ENTER
DATA PRTY STOP BAUD
```



Depress F4 (BAUD)

Select 9600 (make it blink) with F1 (←) to F4 (↓). and depress □.

Depress F2 (PRTY).

Select None (make it blink) with F1 (←) to F4 (↓). and depress □.

Depress F3 (STOP) and F1 (STOP).

Select 1 Bit (make it blink) with F1 (←) to F4 (↓). and depress □.

Depress □ F1.

Select 8 Bits (make it blink) with F1 (←) to F4 (↓). and depress ENTER.

Display

```
Std Mode
OK 04
Select        F No.
Prg + Edit  Trn + Set
```

(2) Transmission Format Setting

Selection of transmission format setting.

SETTING END

Display

```
Std Mode
PKW-3100 Ver ×.××
Select      F No.
Prg + Edit  Trn + Set
```

Depress [F3] (Trn), [□], and [F3] (FMT)

Select INTEL HEX (make it blink) with [←] to [↓] [F4], and depress [ENTER]

Display

```
Std Mode
OK 04
Select      F No.
Prg + Edit  Trn + Set
```

(3) Cable Connection (RS232 C connection)

PROM Writer Side		P150 Side	
Pin No.	Signals	Pin No.	Signals
1	GND	1	GND
2	TXD	2	TXD
3	RXD	3	RXD
4	RTS	4	RTS
5	CTS	5	CTS
6	DSR	6	DSR
7	SG	7	SG
20	DTR	20	DTR

(PKW−3100)

AVAL PECKER 30 ROM Writing

ROM type setting
Automatic selection

SETTING END

With the ROM mounted in the socket,

depress (DEVICE) and (ENTER) .

After confirming the display, depress (ENTER)

*** If automatic selection is not possible, perform the setting by manual as follows:**

ROM type setting
Manual

Maker name setting

Device name setting

①

Display

```
Std Mode
PKW-3100 Ver ×.××
Select        F No.
Prg + Edit  Trn + Set
```

Depress (F4) (Set) , (F1) (DEVS) and (F1) (MARK)

Select maker name (e.g. Hitachi)

(make it blink) with (F1) (←) to (F4) (↓) , and depress (□)

Depress (F1) (DEV)

Select device name (e.g. 27C256H)

(make it blink) with (F1) (←) to (F4) (↓) , and depress (□)

①

Capacity setting

Depress [F1] (SIZE)

Select device name (e.g. 3kB)

(make it bilnk) with [F1] (←) to [F4] (↓),

and depress [□].

Writing method setting

Depress [F1] (PROC)

Select device name (e.g. FUJITSU)

(make it blink) with [F1] (←) to

[F4] (↓) and depress [ENTER].

SETTING END

Internal RAM clear

Display

```
Std Mode
PKW-3100 Ver ×.××
Select       F No.
Prg + Edit   Trn + Set
```

Depress [F2] (Edit) [F2] (CLB) [F2] (ALL)

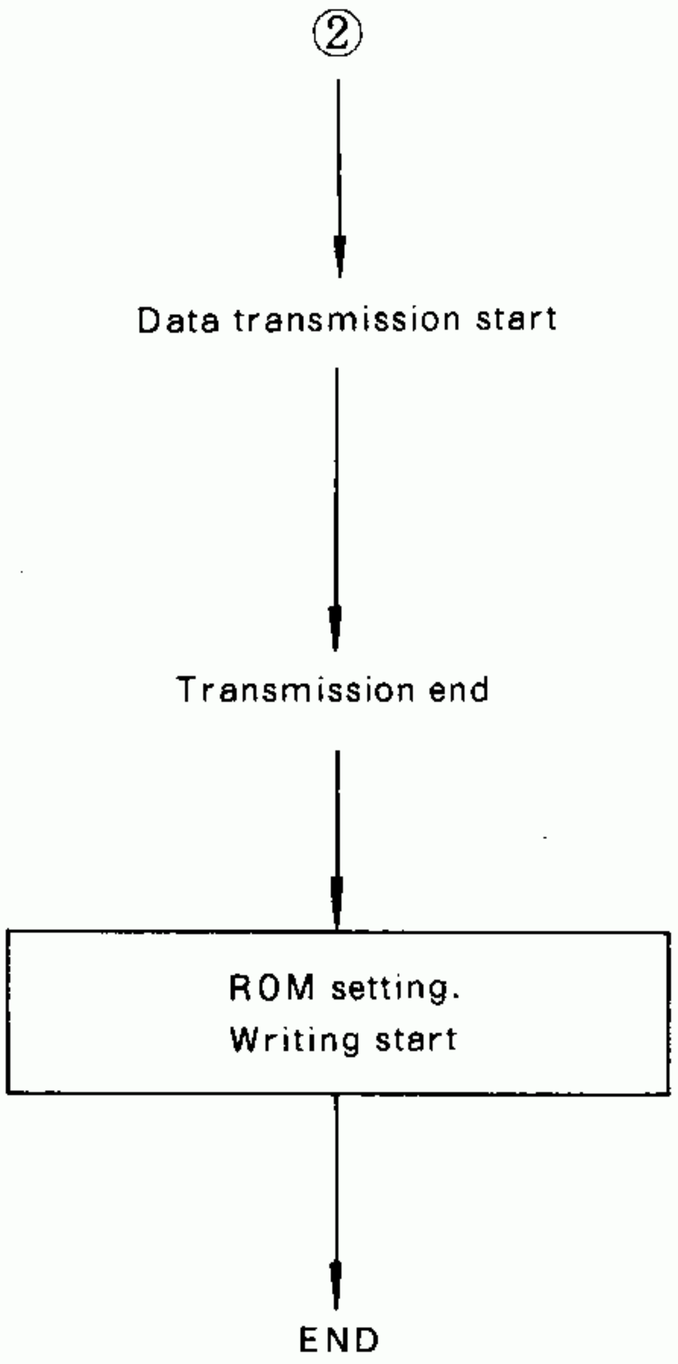
and [ENTER]

Depress [F3] (Trn) [F1] (RRS) [F2] (ALL)

and [ENTER]

Data receiving

②



Start transmission from the P150 side in sequence of

F→ EPROM → File selection → CONFIRM .

Depress PROG and ENTER .

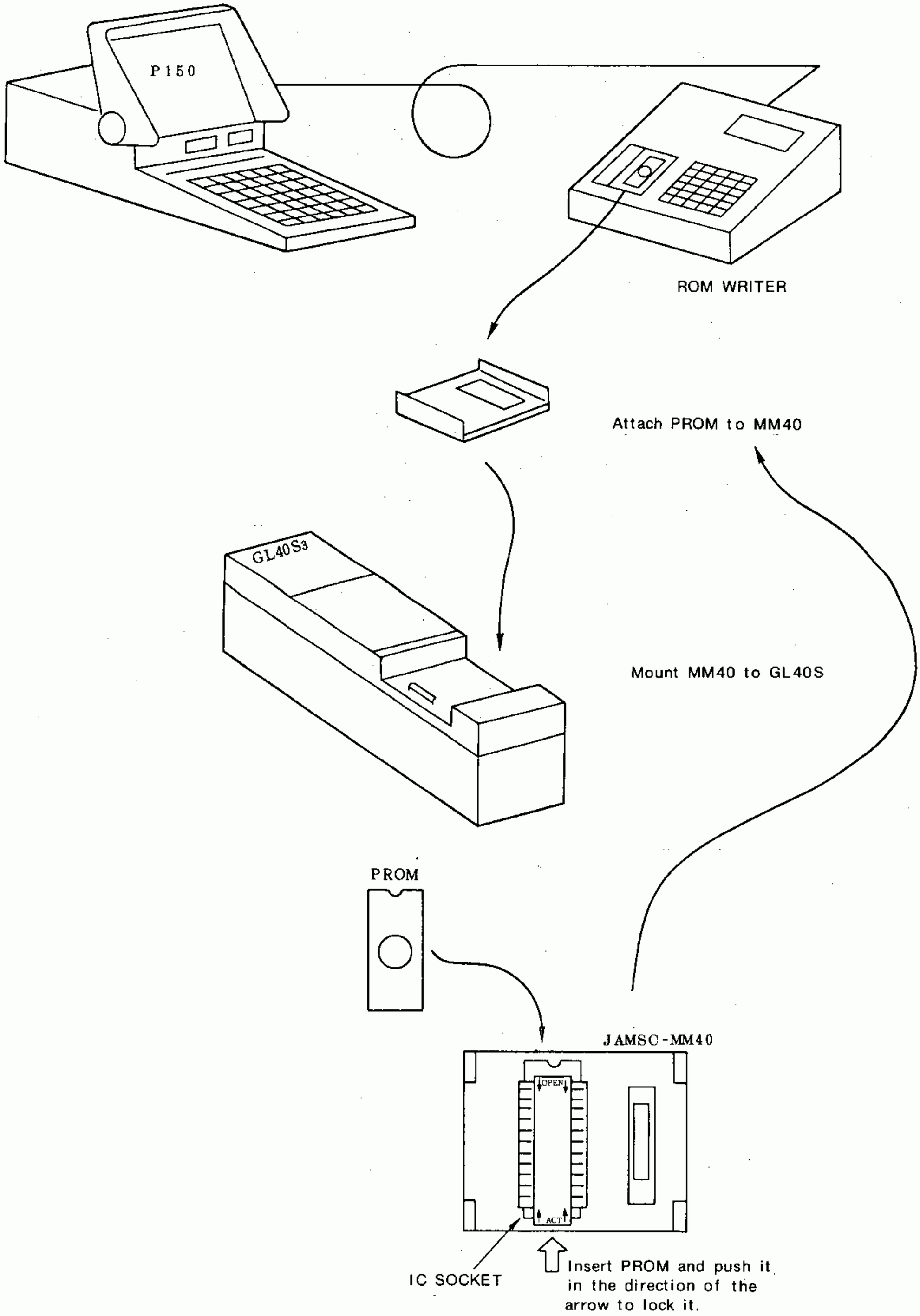
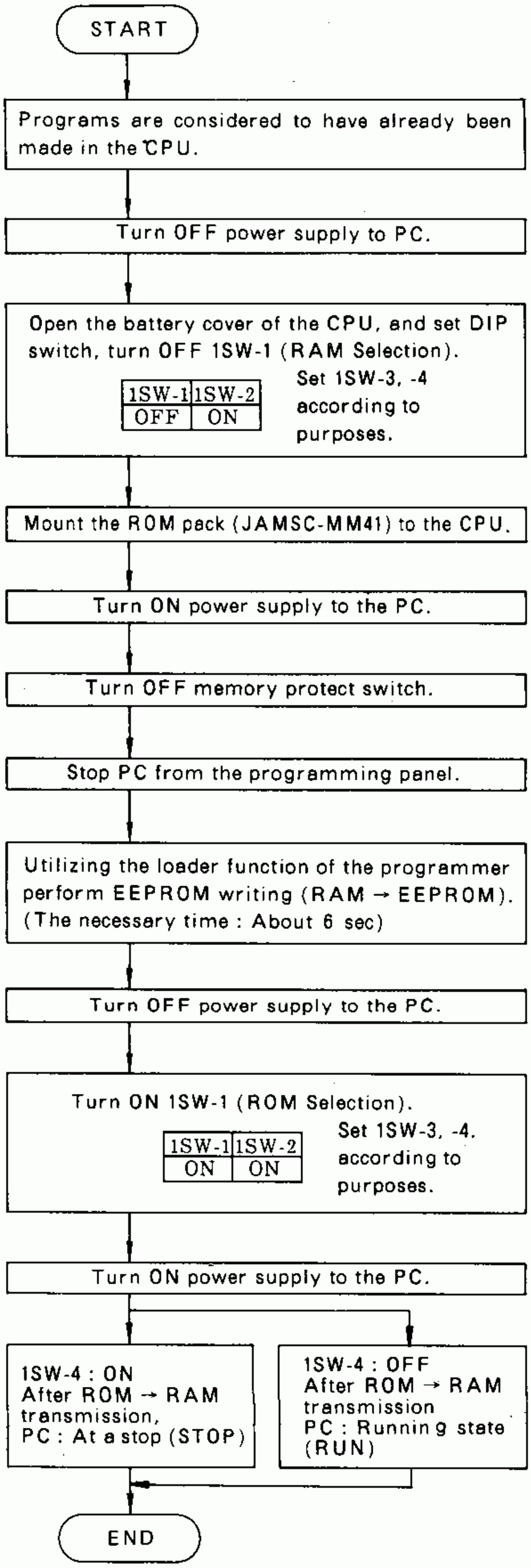


Fig. 7.2 ROM Pack Writing Scheme

7.2.3 EEPROM

The programs of GL40S are written in an EEPROM through the operation of the programming panel.

EEPROM Operation



Note

1. If ROM→RAM transmission is not performed at the time of power ON, the following cases may occur.

   (1) Data in the ROM is not correct.
   (2) The size of program data in the ROM is larger than the memory size in the CPU. For example, programs are written in the ROM up to 3k, while the CPU mounted is of 2k.
   (3) The ROM is not mounted.
   (4) Switch setting is not ROM operation.

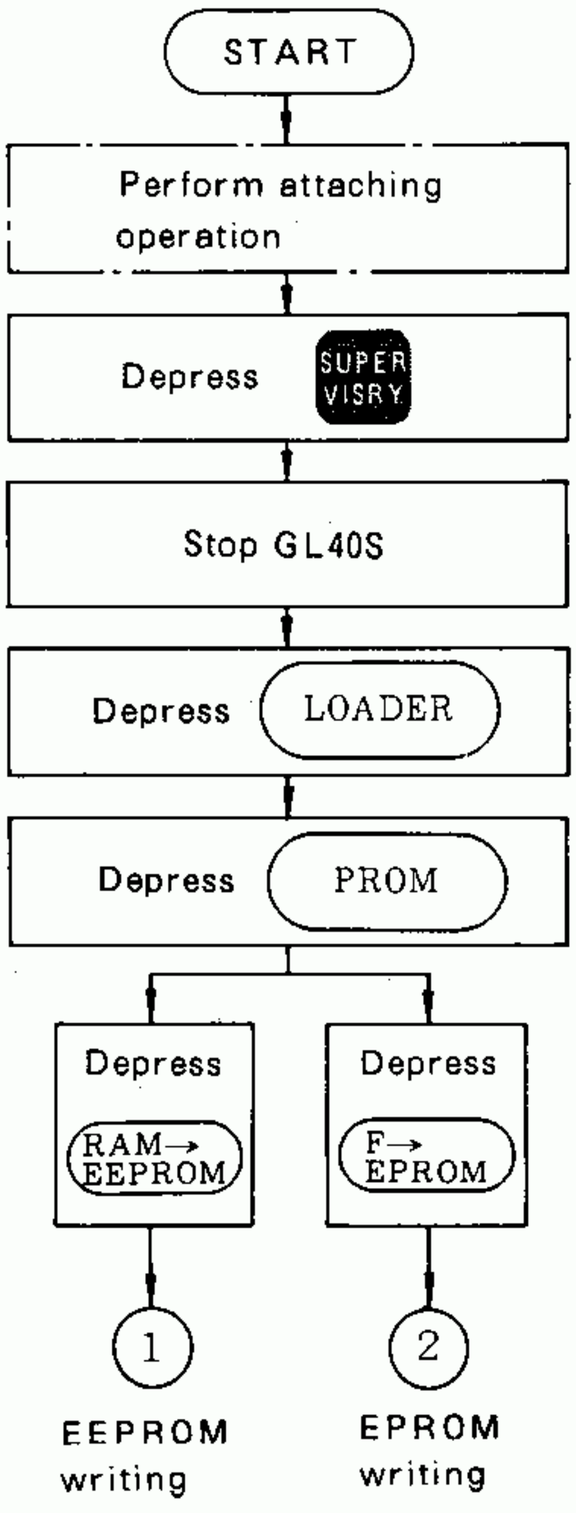
2. In case of start operation from the programming panel, ROM→RAM transmission is not performed.

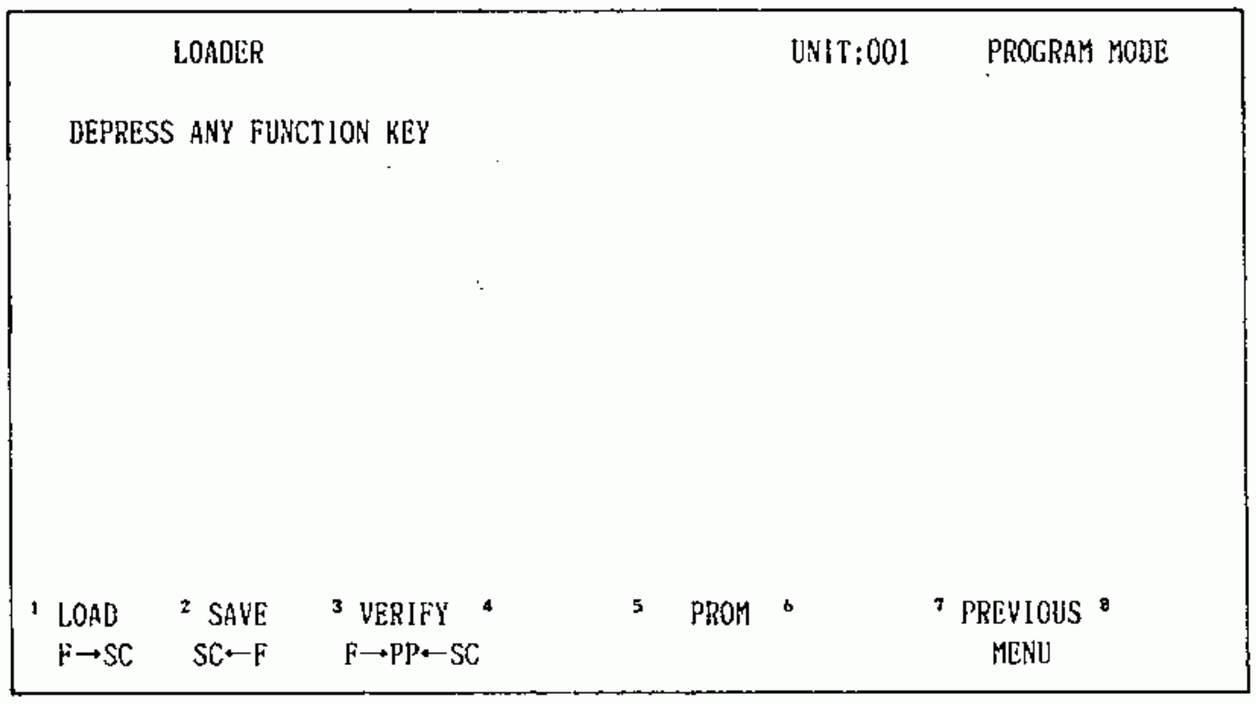
3. Irrespective of the state of the memory protect switch, ROM→RAM transmission is performed.

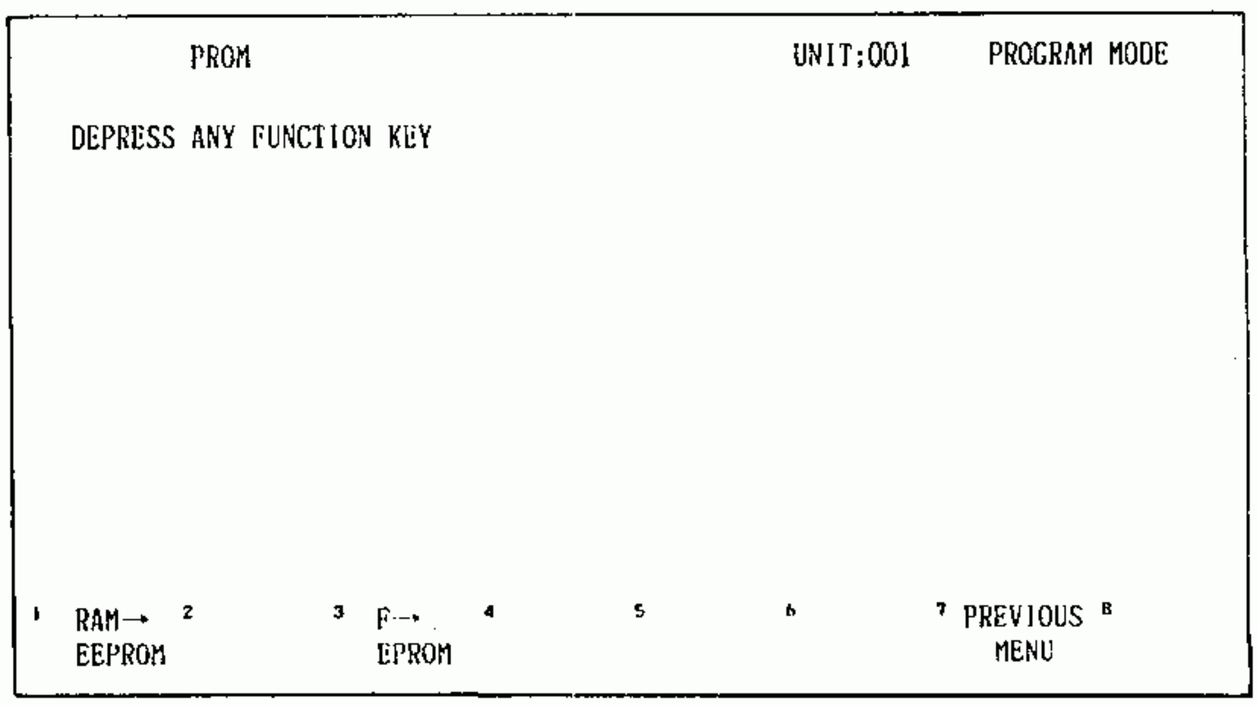
7.3 ROM MAKING

PROM Making

The operation for writing programs in the EEPROM and EPROM. EPROM writing requires a disk in which programs are written.



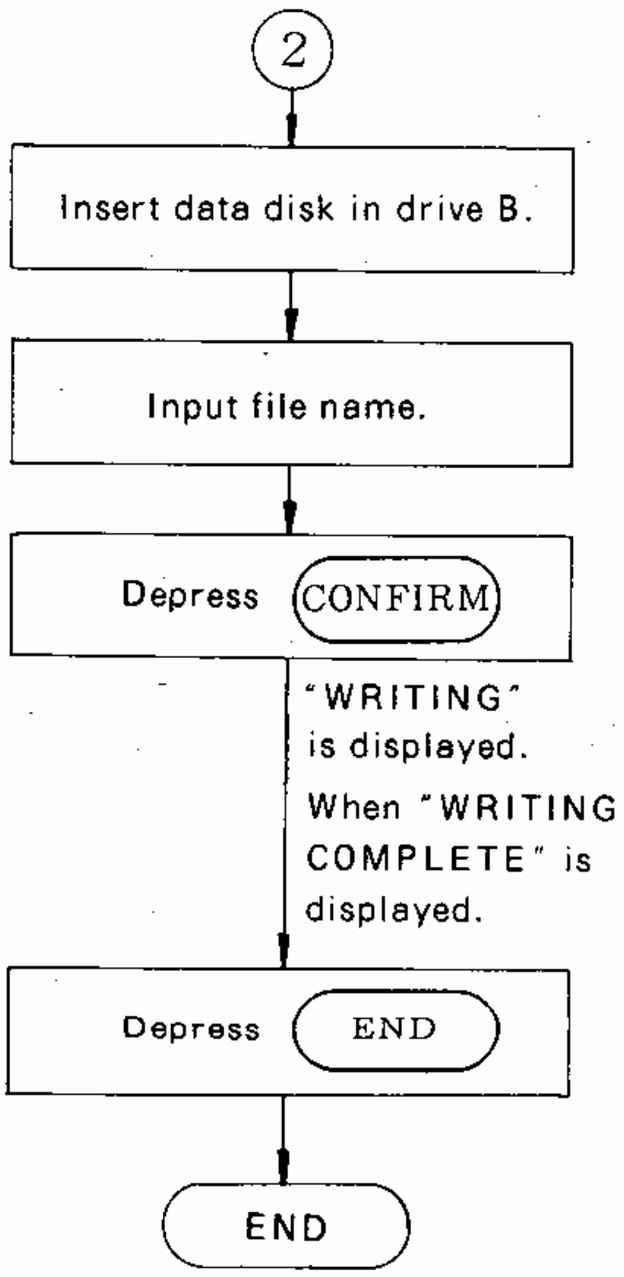




(Comment)

For EEPROM writing, attach MM41 to the EEPROM beforehand.

EPROM (MM40) Writing




```
        F→EPROM                                  UNIT;001     PROGRAM MODE

  INSERT DATA DISK IN DRIVE B :  AND INPUT FILE NAME
  FILE NAME :




  WRITE REQUESTED

 1DIRECTORY2        3         4         5         6         7         8 CANCEL
```

```
        F→EPROM                                  UNIT;001     PROGRAM MODE

  FILE NAME : B:GL40S
  TITLE     : MEMOCON-SC GL40S
  DATA      : 11-06-1989
  ORDER#    : 12345




  WRITE REQUESTED

 1 CONFIRM 2         3         4         5         6         7         8 CANCEL
```

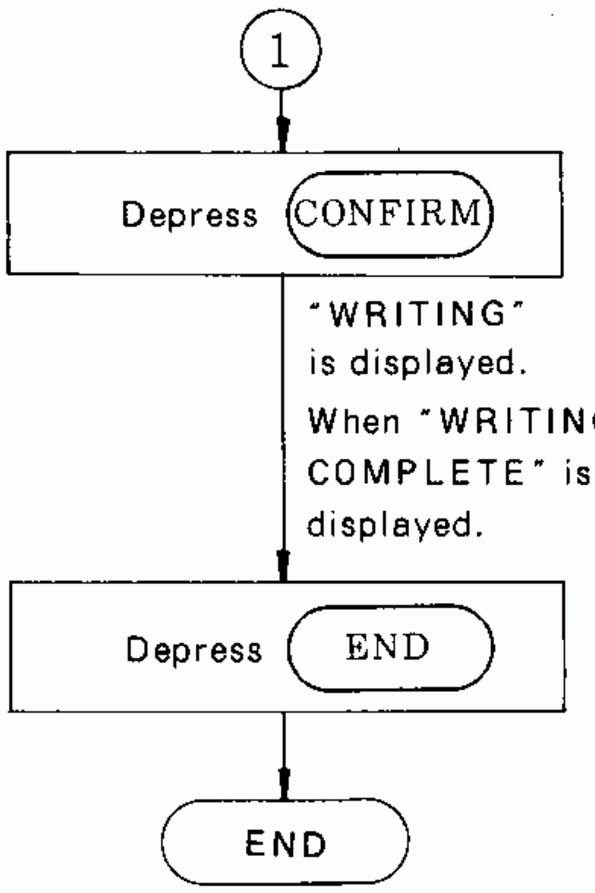
```
        F→EPROM                                  UNIT;001     PROGRAM MODE

  FILE NAME : B:GL40S
  TITLE     : MEMOCON-SC GL40S
  DATA      : 11-06-1989
  ORDER#    : 12345




 1         2 END     3         4         5         6         7         8
```

EEPROM (MM41) Writing




```
     RAM → EEPROM                                  UNIT:001     PROGRAM MODE
  DEPRESS ANY FUNCTION KEY




1          2  END  3          4          5          6          7          8
```

```
     RAM → EEPROM                                  UNIT:001     PROGRAM MODE
  DEPRESS ANY FUNCTION KEY




  WRITE REQUESTED

1 CONFIRM 2          3          4          5          6          7          8 CANCEL
```

SECTION 8
GL40S HANDLING

8.1 NOTES ON HANDLING

Memocon-SC GL40S should be used to meet your system specifications paying attention to the following points.

8.1.1 Backup Circuit

Since GL40S is more reliable than relays, and their eventual faults can be repaired quickly, it requires no backup circuit in ordinary systems. However, if it does require a backup circuit because of the special nature of the system, the selection of a proper backup method is an important consideration. An external manual circuit and standby GL40S are sound methods.

8.1.2 Interlock

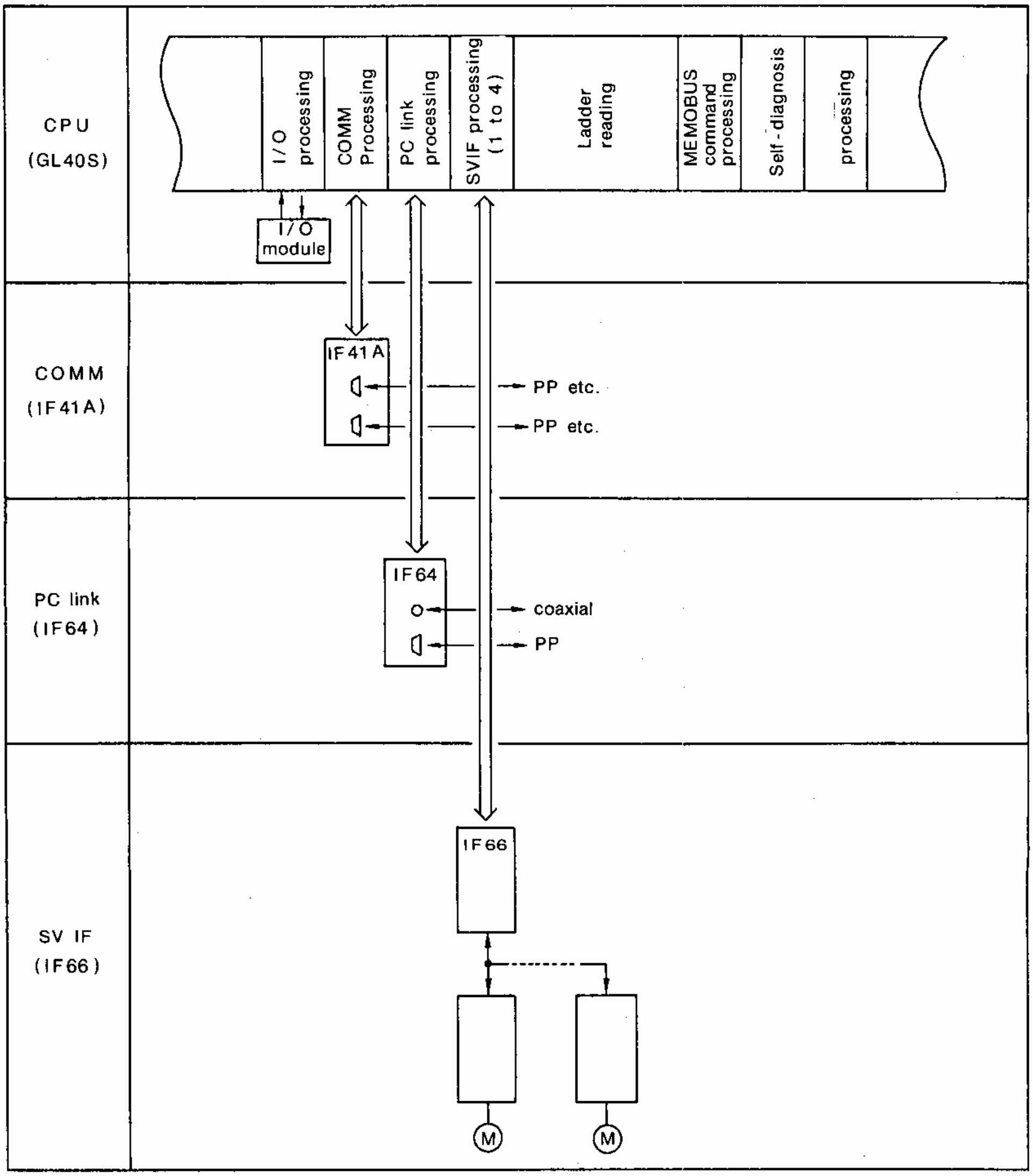
The GL40S CPU is provided with a self-diagnosis function of stopping operation and turning output OFF when the stored ladder circuits (program) are destroyed, or when module cards develop faults. However, some faults and misoperations are not detected, and may destroy machines and devices. If there is a possibility of such destruction, start and stop the system under redundant control such as external interlock and electrical-mechanical redundant control. Refer to Par. 8.4.2 (3), "Power Supply Circuit".

8.1.3 Control Panel Layout

Lay out the control panel locations in consideration of the electrical environment conditions. For details, refer to Par. 8.4 "CONSTRUCTION, INSTALLATION AND WIRING OF CONTROL PANEL."

8.1.4 I/O Service

The I/O processing of GL40S is performed as follows. Ladder reading and processings of COMM module, PC link module and servo I/F module are done in parallel.



8.1.4 I/O Service (Cont'd)

The network processing method of GL40S has been explained in this manual. However, for reading the ON and OFF states of all input signals into the CPU module of GL40S correctly, the ON and OFF states of input signals must continue longer than the total delay time in the input module and one scan time. Therefore, when dealing with signals of short duration, special devices such as external memory circuits must be used, and for limit switch signals, the dog length must be sufficiently long. Refer to Fig. 8.1.

Output signals are also delayed up to the total of one scan time and the delay time within the output module. Therefore, for applications requiring a high degree of accuracy, some external devices are required. Refer to Fig. 8.2.

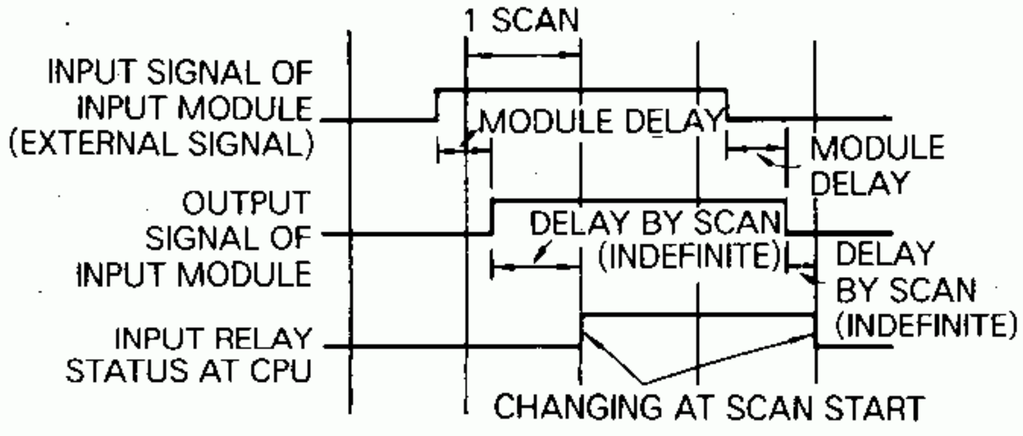


Fig. 8.1 Input Signal Delay

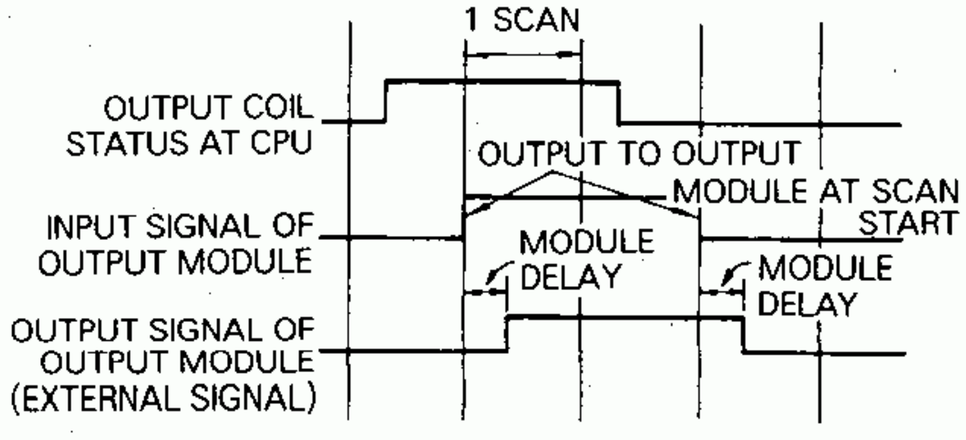


Fig. 8.2 Output Signal Delay

8.2 CALCULATION OF MEMORY CAPACITY

To find the required number of memories is a difficult task in composing any system. Exact numbers can only be determined from the intended ladder circuits, but here, the memory capacity of GL40S may be roughly found as follows. Although more memories are required when the number of I/O signals increases, and when the sequence becomes more complex (more complex control),

$$40 \times \text{Number of Output Coils}$$

is taken as a rough guideline regarding the number of memories (words).

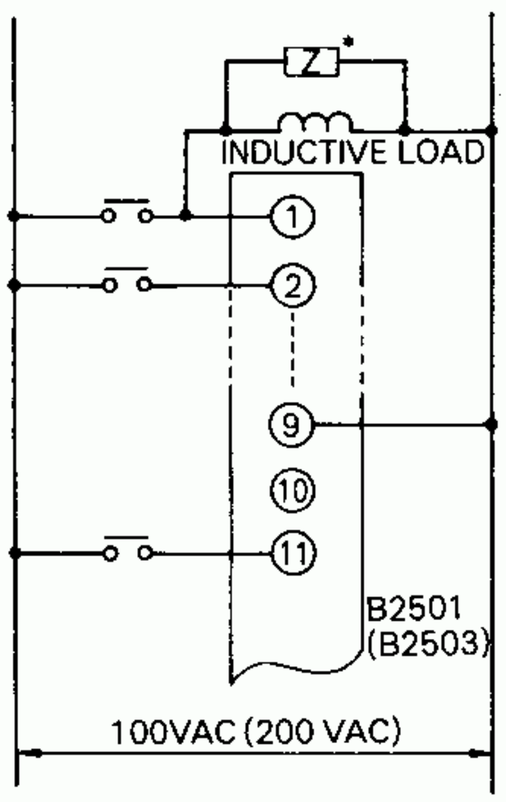
Additional sequences may become required during trial and adjustment operations, and for this reason, some reserve memories should be prepared from the beginning.

8.3 PRECAUTIONS FOR USING I/O MODULES

8.3.1 Input Module

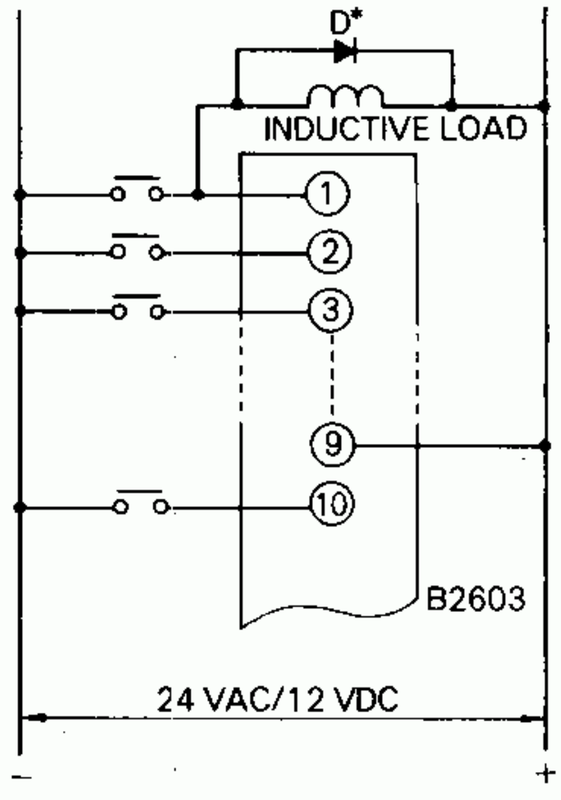
(1) Inductive Load

Where an inductive load is connected in parallel with the input module as shown in Figs. 8.3 and 8.4, connect a surge absorber or flywheel diode in parallel with the inductive load, respectively for the AC input module and DC input module.



*The surge absorber capacity should be selected corresponding to the load. It is recommeded that type CR 50500 (made by Okaya Electric Industries Co.) or equivalent be used.

Fig. 8.3 AC Input Module

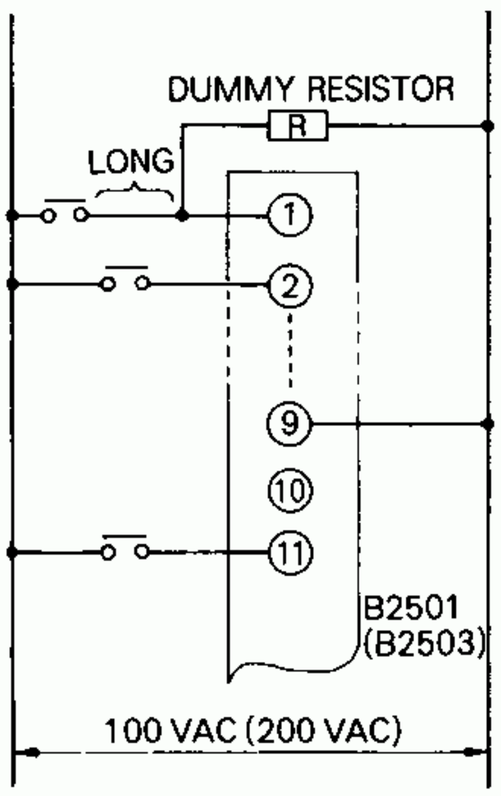


*The flywheel diode should be selected corresponding to the load. It is recommended that type F14 series (made by NEC) or equivalent be used.

Fig. 8.4 DC Input Module

(2) Input Dummy Resistor

Where the external wiring is long or where there is an induction source in the vicinity, connect a dummy resistor in parallel to the input module, as shown in Fig. 8.5.



R : input dummy resistor
B2501 : 5 kΩ (10 W or more)
B2503 : 10 kΩ (20 W or more)

Fig. 8.5 AC Input Module

(3) Leakage Current in Input Equipment

Some input equipment (e.g. noncontact switches and limit swithces with LED) has leakage current during the OFF state. If this equipment is connected to AC input modules, it may fail to maintain the voltage for an OFF condition which is an input due to leakage current, and input signals may not be cut off.

(Example) A non-contact switch with 5 mA of leakage current is connected to B2501.

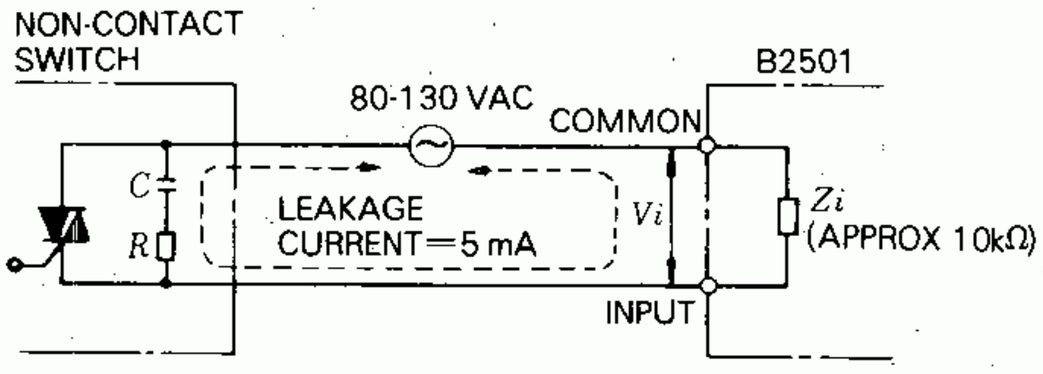


Fig. 8.6 Connection of a Non-contact Switch

If the leakage current is 5 mA, the input voltage (Vi) of B2501 becomes;

$$Vi = 5mA \times Zi = 5mA \times 10k\,\Omega = 50V$$

Since this does not satisfy an input condition (OFF voltage = 30 V or less), input signals may not be cut off. In this case, add a proper dummy resistor to the input terminal of B2501.

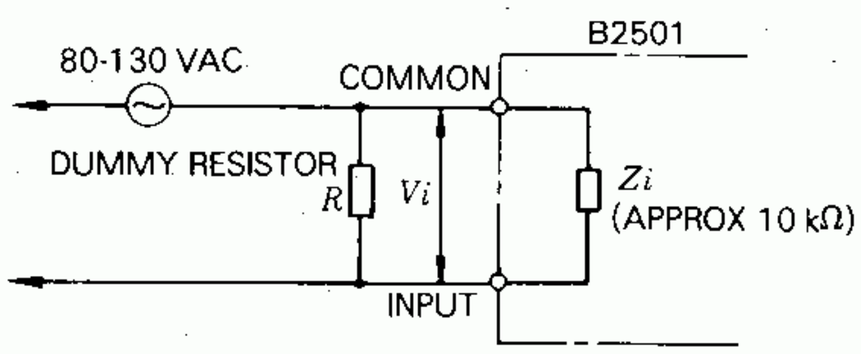


Fig. 8.7 Addition of a Dummy Resistor

The value of the dummy resistor R should be decided so that an input voltage Vi of B2501 becomes 30 V or less.

$$\frac{R \times Zi}{R + Zi} \times leakage\ current < 30\,V$$

$$\frac{R \times 10k\,\Omega}{R + 10k\,\Omega} \times 5mA < 30\,V \qquad \therefore R < 15k\,\Omega$$

Thus, the value of R becomes 15 kΩ or less. However, if the value is too small, heating value increases, resulting in the need of a resistor with large wattage.

Assume that the value of R is 10 kΩ. Then the wattage W of the dummy resistor becomes;

$$W = \frac{(power\ source)^2}{R} = \frac{(100\,V)^2}{10k\,\Omega} = 1\,W \qquad \therefore W = 1\,W$$

Generally, the wattage W of the dummy resistor is taken to be 3W to provide a surplus wattage about three times more than required.

(4) ON/OFF Conditions of DC Input Module

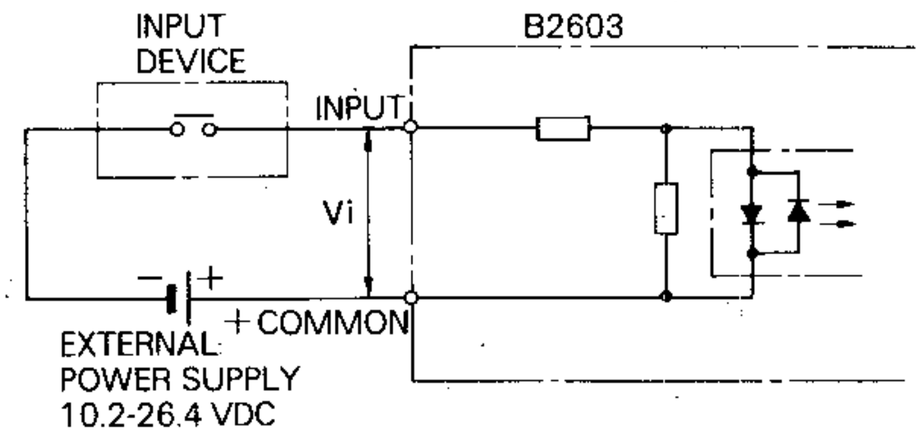


Fig. 8.8 ON/OFF Conditions of B2603

Input conditions of B2603 are;
ON level: 9 VDC or more
OFF level: 6 VDC or less

(Example) When a limit switch with LED is connected to B2603.

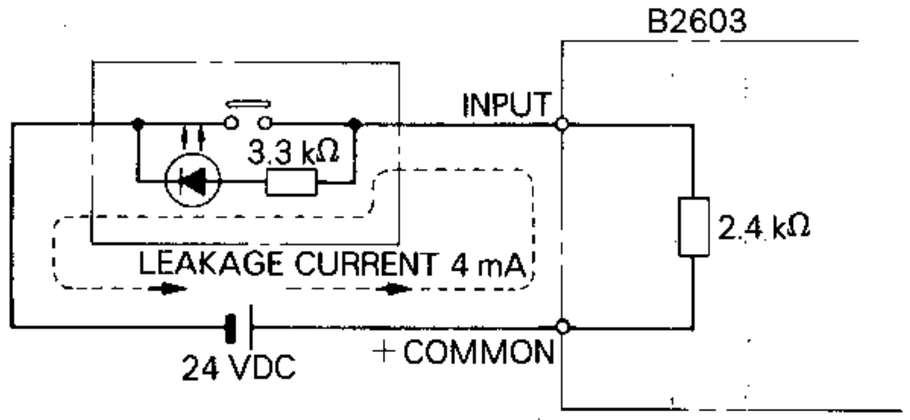


Fig. 8.9 Connection of a Limit Switch with LED

If the leakage current is 4 mA, then

$$Vi = 2.4\,k\Omega \times 4mA = 9.6\,V$$

This does not satisfy the input condition (OFF level = 6 V or less). Therefore, add a proper dummy resistor to the input terminal of B2603 so that the input condition is satisfied.

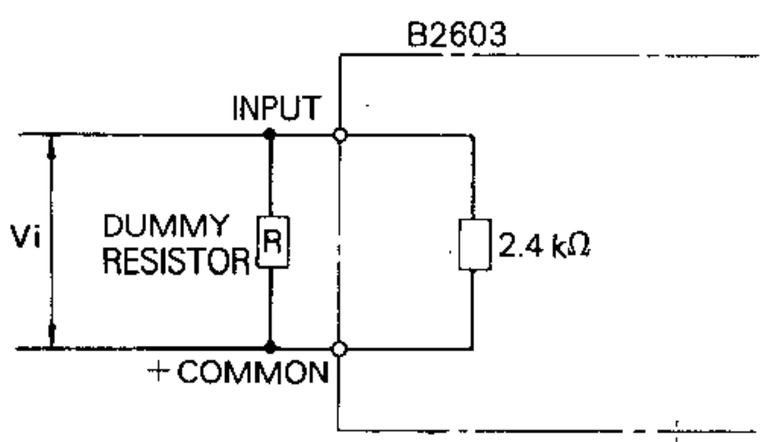


Fig. 8.10 Addition of a Dummy Resistor

The value of a dummy resistor R should be chosen such that the input voltage V_I of B2603 becomes 6 V or less.

$$4mA \times \frac{2.4k\Omega \times R}{2.4k\Omega + R} < 6V \qquad \therefore R < 4k\Omega$$

Thus, the value of the resistor becomes 3 kΩ.
Necessary wattage W is;

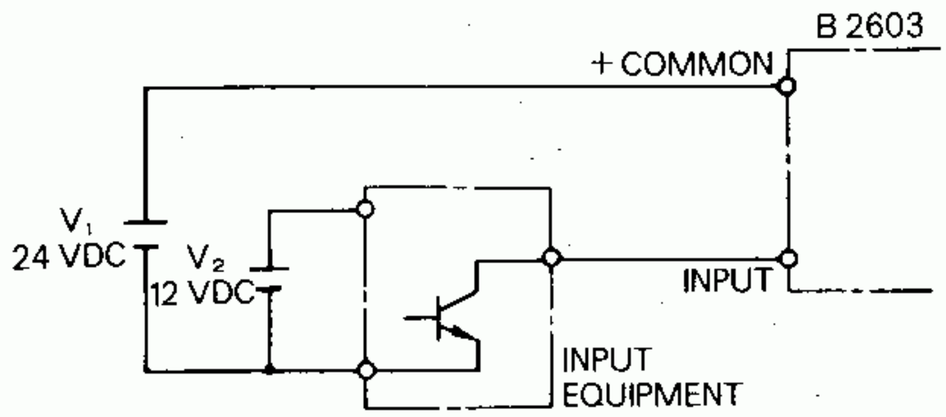
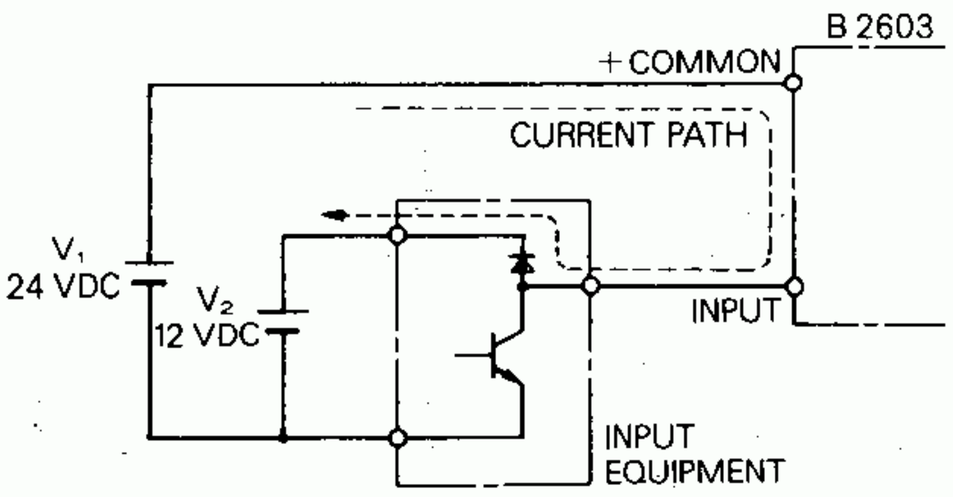
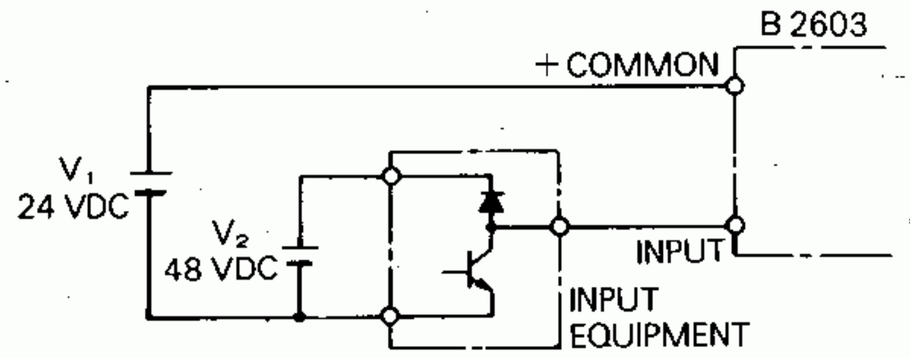
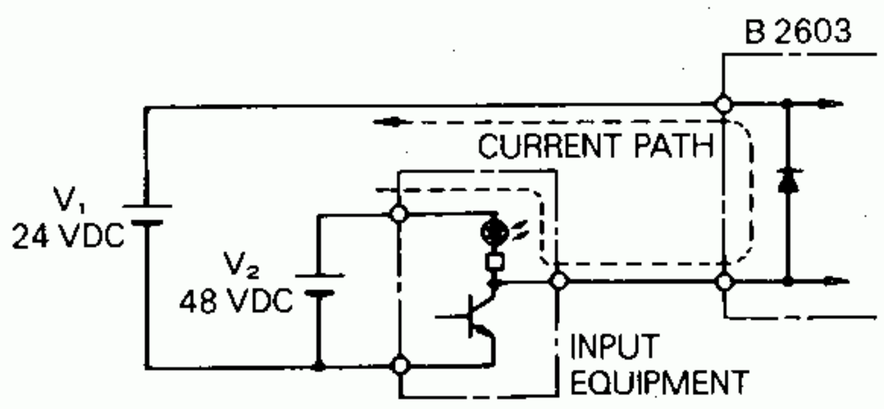
$$W = \frac{(power\ supply\ voltage)^2}{R} = \frac{(24\,V)^2}{3k\Omega} = about\ 200\ mW$$

In general, the wattage of a dummy resistor is taken to be 0.5-1 W to provide a surplus wattage about three times more than required.

(5) Connection to Input Equipment with Different Voltage

Usually, power voltage of input equipment should be matched that of input modules. However, Table 8.1 shows possibilities of connecting input equipment having different voltages.

Table 8.1 Possibilities of Connecting Input Equipment Having Different Voltages

Example of Input Equipment	Connection Possibilities
① Open collector output ($V_1 > V_2$) 	Can be connected. However, the voltage resistance of the output transistor of the input equipment should be 40 V or more.
② With resistor, LED or diode ($V_1 > V_2$) 	Cannot be connected. When the input equipment is OFF, current shown by a dotted line in the left figure may flow and input does not become OFF. Especially, in case of LED, reverse voltage may be applied during the OFF time to the equipment with LED and the LED may be broken.
③ With open collector or diode ($V_1 < V_2$) 	Can be connected.
④ With resistor or LED ($V_1 < V_2$) 	Cannot be connected. When the input equipment is OFF, current shown by a dotted line in the left figure may flow and the LED of the input equipment comes on dimly.

(6) Caution in Using B2603

Ambient temperature of B2603 (32-point) input modules depends on the external power supply voltage and the number of input points that are ON at the same time. Adjust the ambient temperature within a value shown in Fig. 8.11.

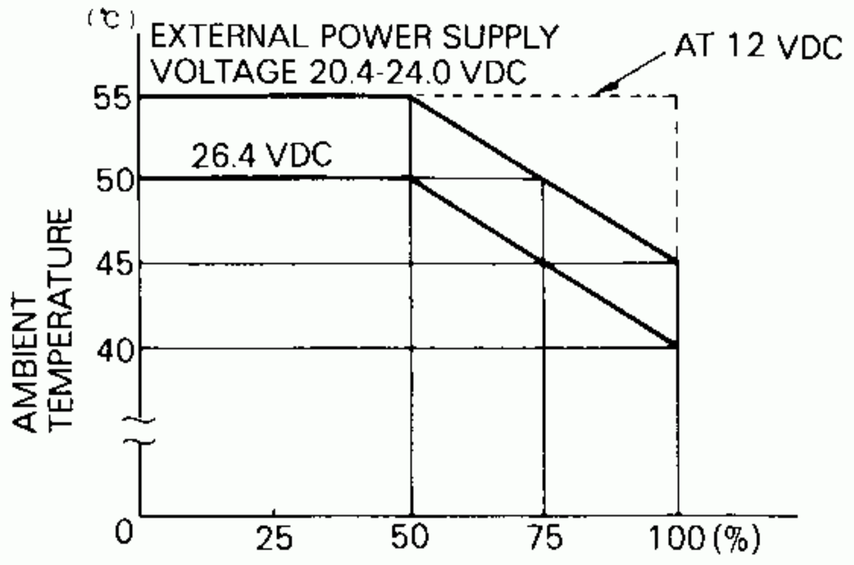
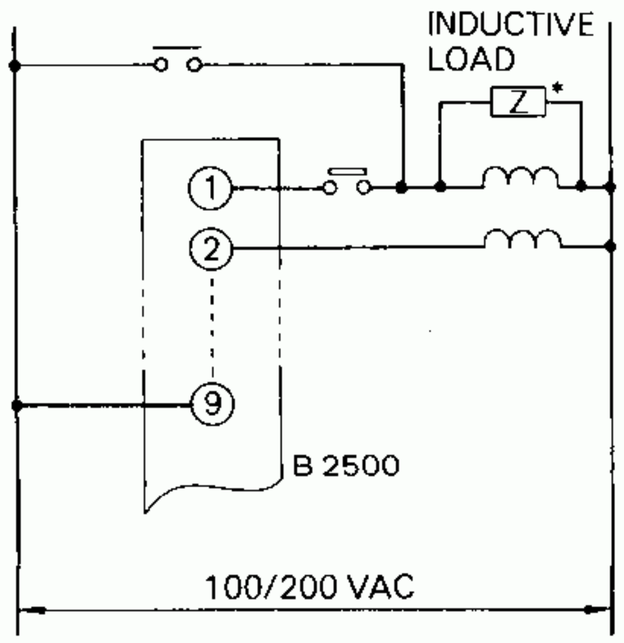


Fig. 8.11 Adjustment of Ambient Temperature

8.3.2 Output Module

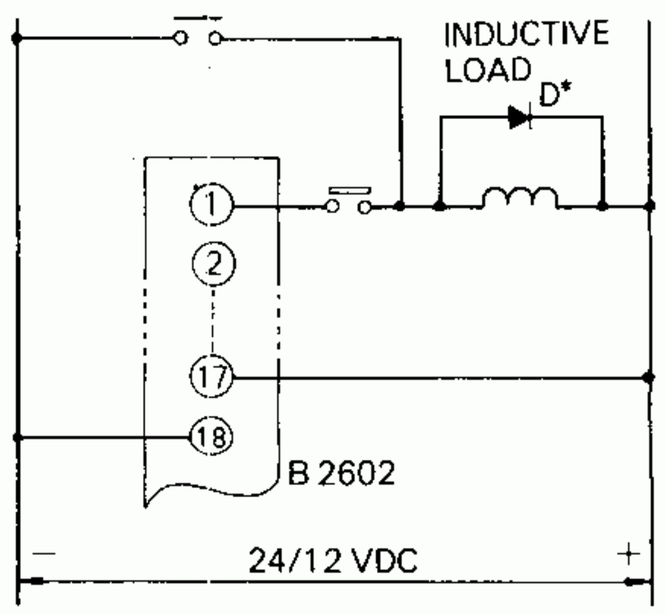
(1) Connection to Contacts

Where connecting contacts to an inductive load of the output module, as shown in Figs. 8.12 and 8.13, always connect a surge absorber or a flywheel diode in parallel to the inductive load.



*The surge absorber capacity sould be selected corresponding to the load. It is recommended that type CR 50500 (made by Okaya Electric Industries Co.) or equivalent be used.

Fig. 8.12 AC Output Module



*The flywheel diode should be selected corresponding to the load. It is recommedned that type F14 series (made by NEC) or equivalent be used.

Fig. 8.13 DC Output Module

(2) Minimum Load Current

As the output switch of the AC output module, a triac is used. Since a triac cannot operate stably if the load is less than the specified minimum load current, make sure to use the load which is secure current levels above the minimum load current. If the minimum load current cannot be kept, connect a dummy resistor in parallel to the load so that the total load current is above the minimum load current.

(3) Maximum Load Current

For B2500, for example, although an output point can accommodate a 1 A load, the total load for 8 output points must be up to 3 A. This should be taken into consideration for distributing loads.

(4) Output Fuse

The output fuse is used for preventing the trouble caused by shortcircuit of the load, but not for protecting the output element of the module.

(5) Status LED Indicator for AC Output Module

The status LED indicator for the AC output module lights up by power supply for the internal logic circuit.

(6) Leakage Current from the Output Module

AC output module and relay contact output module contain a surge surpressing circuit. Therefore, leakage current flows during OFF.

Table 8.2 Leakage Current in Output Modules

Output Module : Type	Output Impedance during OFF (50 Hz)	Maximum Leakage Current
JAMSC-B2500	Approx. 68 kΩ	Approx. 2 mA at 130 VAC

When a light-load relay is connected to these output modules, the relay does not turn off due to the current.

(Example) When load impedance is 6 kΩ and the load responds incorrectly due to 1 mA of leakage current.

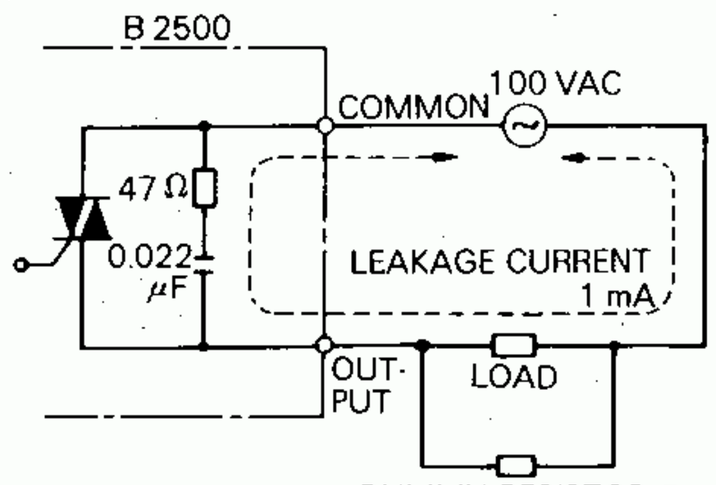


Fig. 8.14 Connection of Light Load

If 0.5 mA or less of current flow in the load does not cause any malfunction, then the value of dummy resistor R becomes;

$$1\,mA \times \frac{R}{R + 6\,k\Omega} < 0.5\,mA$$

$$R < 6\,k\Omega$$

Thus, make the value of R 6 kΩ.

Necessary wattage W is;

$$W = \frac{(power\ source\ voltage)^2}{R} = \frac{(100\,V)^2}{6\,k\Omega} = 1.7\,W$$

Generally, the wattage of the dummy resistor is taken to be 5 W to provide surplus wattage about 3 times more than required.

(7) Connection of Solenoid with Diode

Solenoids with diodes have the advantage being driven by half-wave rectification and less starting current. When solenoids with diodes are used as load of AC output module, be careful of the following points.

① When output is OFF, overvoltage is applied to load:

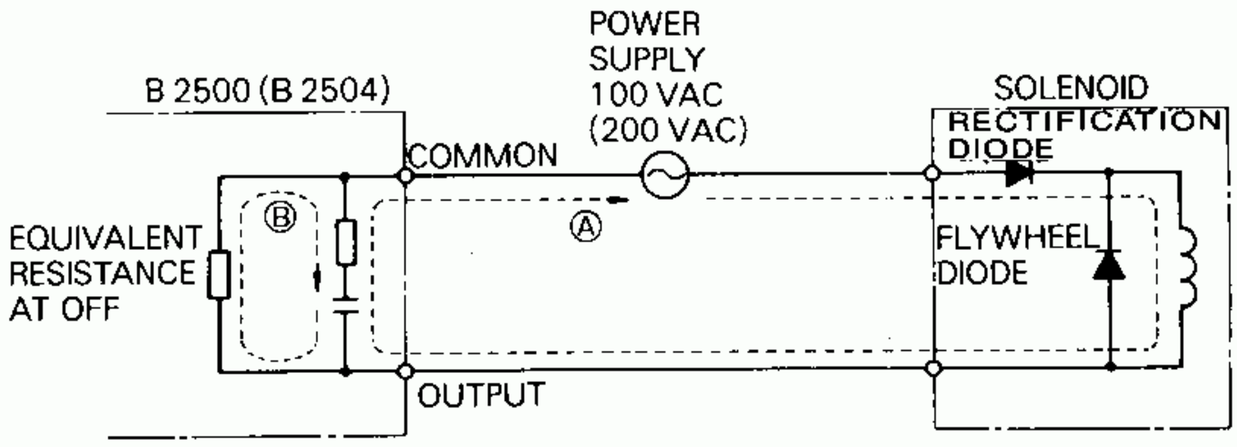


Fig. 8.15 Connection of Solenoid with Diode

When output of AC output module is OFF, current Ⓐ, shown by a dotted line, flows at half-cycle of the power supply to be a forward-biased rectification diode, and is charged on capacitors. See Fig. 8.15.

With next half-wave, after polarity is reveresed, rectification diode is reverse-biased and current Ⓐ is blocked; discharge current Ⓑ , shown by a dotted line, flows from the capacitor. At this time, supply voltage and voltage charged on the capacitor are superimposed, and applied to the solenoid. The peak value of this voltage is approximately $2\sqrt{2E}$ (E: supply voltage). Rectification diode should require a withstand reverse voltage of $2\sqrt{2E}$ or more.

Connect a resistance of approximate multiples of 10 k Ω to several hundred k Ω on solenoid ends so that voltage applied to the solenoid with rectification diode is reduced. See Fig. 8.16.

② When output is ON, solenoid may not turn ON:

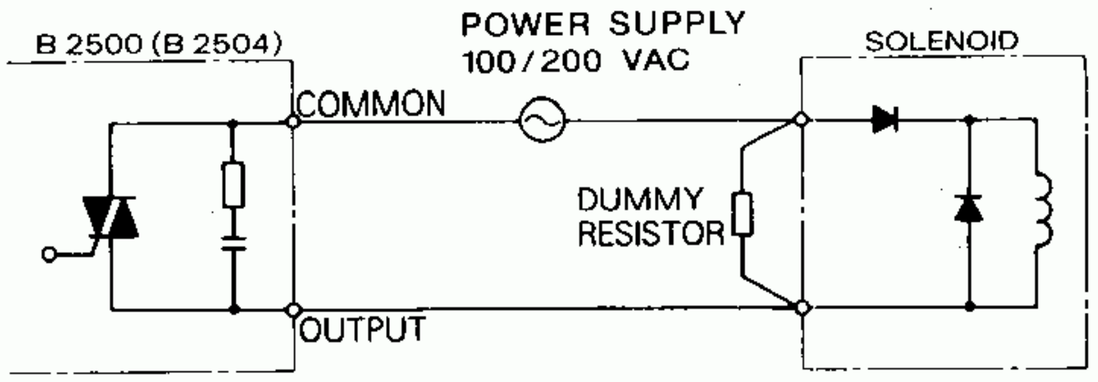


Fig. 8.16 Connection of Dummy Resister

Where the solenoid with diode is connected, solenoid may not turn ON because voltage of output ends is not reduced to operation level by effect of voltage charged in the capacitor. Connect resistance of approximate multiples of 10 k Ω to several hundred k Ω on solenoid ends so that voltage applied to solenoid is reduced. See Fig. 8.16.

(8) Connection of B2904, B2914

Bestact relay output module, B2904, B2914 have a polarity on contact output as shown in Fig. 8.17. When DC power supply is used as a load, observe the correct polarity. If using an opposite polarity, electrical life of the contact may be shortened.

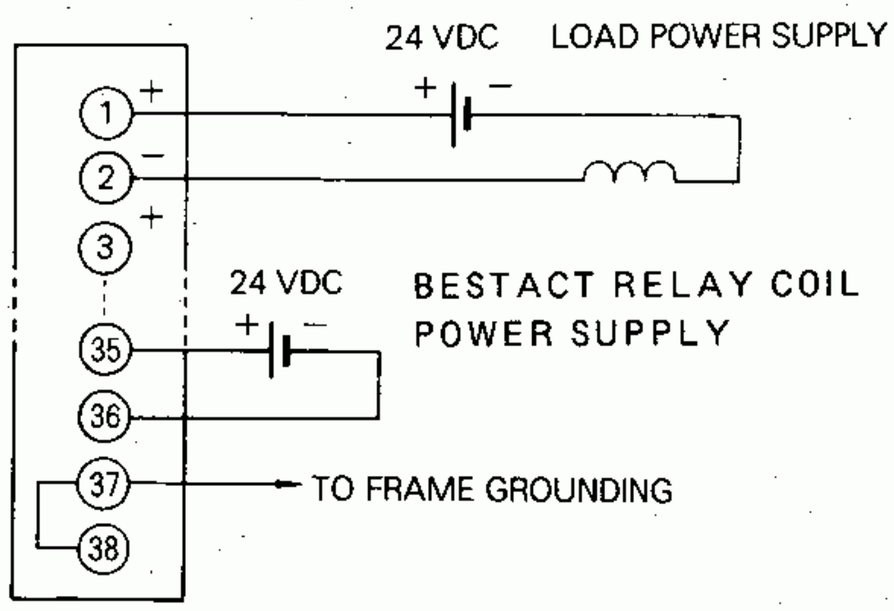


Fig. 8.17 Connection of B2904 and DC Load

8.3.3 Connection between I/O Modules

Where two or more GL40S controllers are used in a system, and signals are exchanged between GL40S controllers via I/O modules, connections should be as shown in Figs. 8.18 and 8.19. In this case, make sure to use modules of the same voltage rating, and connect a dummy resistor to the output module, to make stable operation.

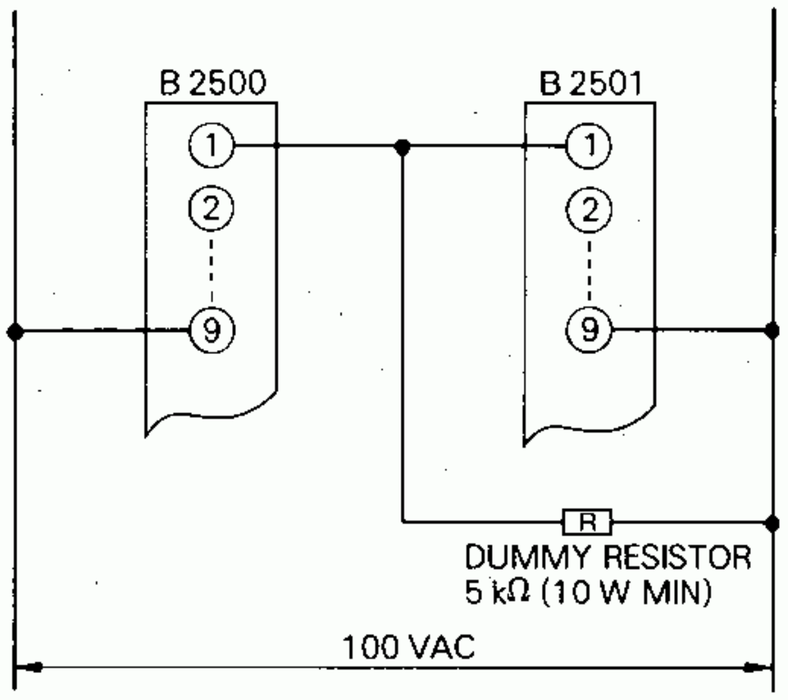


Fig. 8.18 Connection between AC I/O Modules

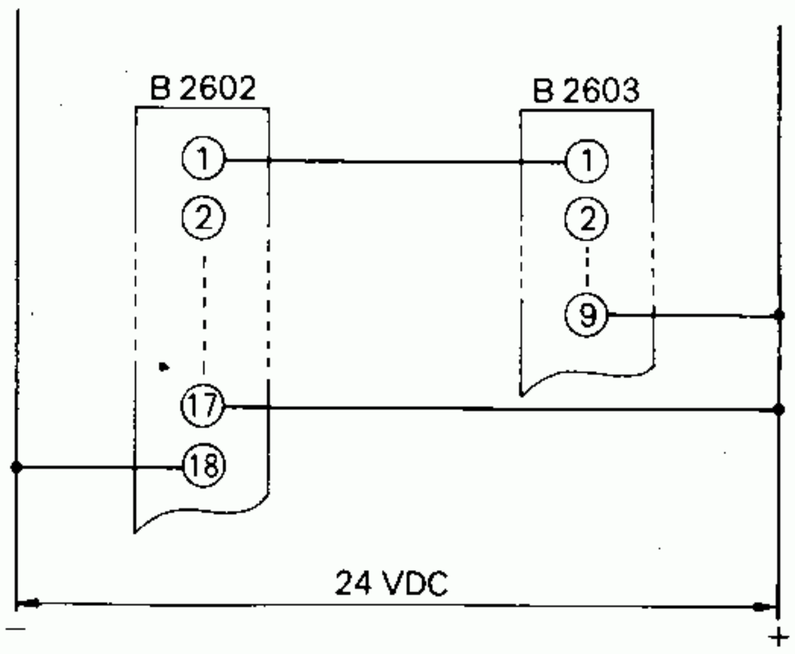


Fig. 8.19 Connection between DC I/O Modules

8.3.4 External Power Supply

General DC stabilized power supply should be used for DC I/O modules as an external power supply. Add a noise filter on the AC input side of the DC stabilized power supply for special modules such as a register module, analog module, or counter module. Do not run the primary and the secondary side of the noise filter and the DC output side in the same wire duct.

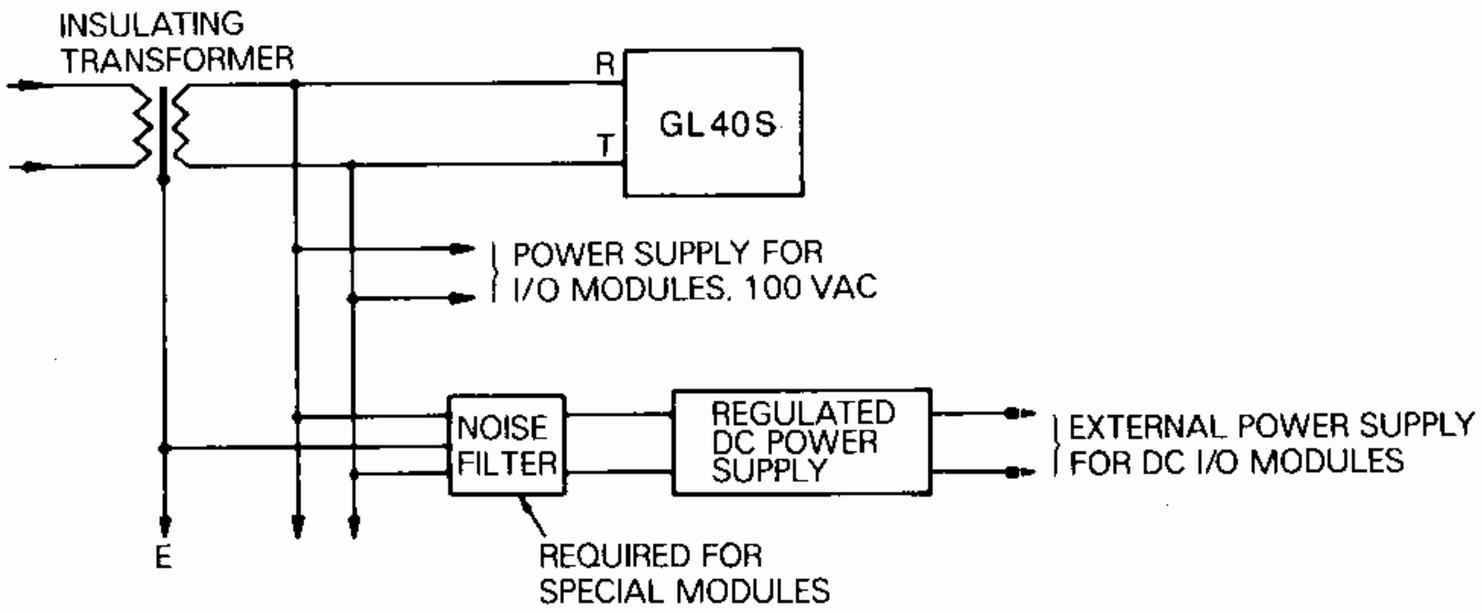


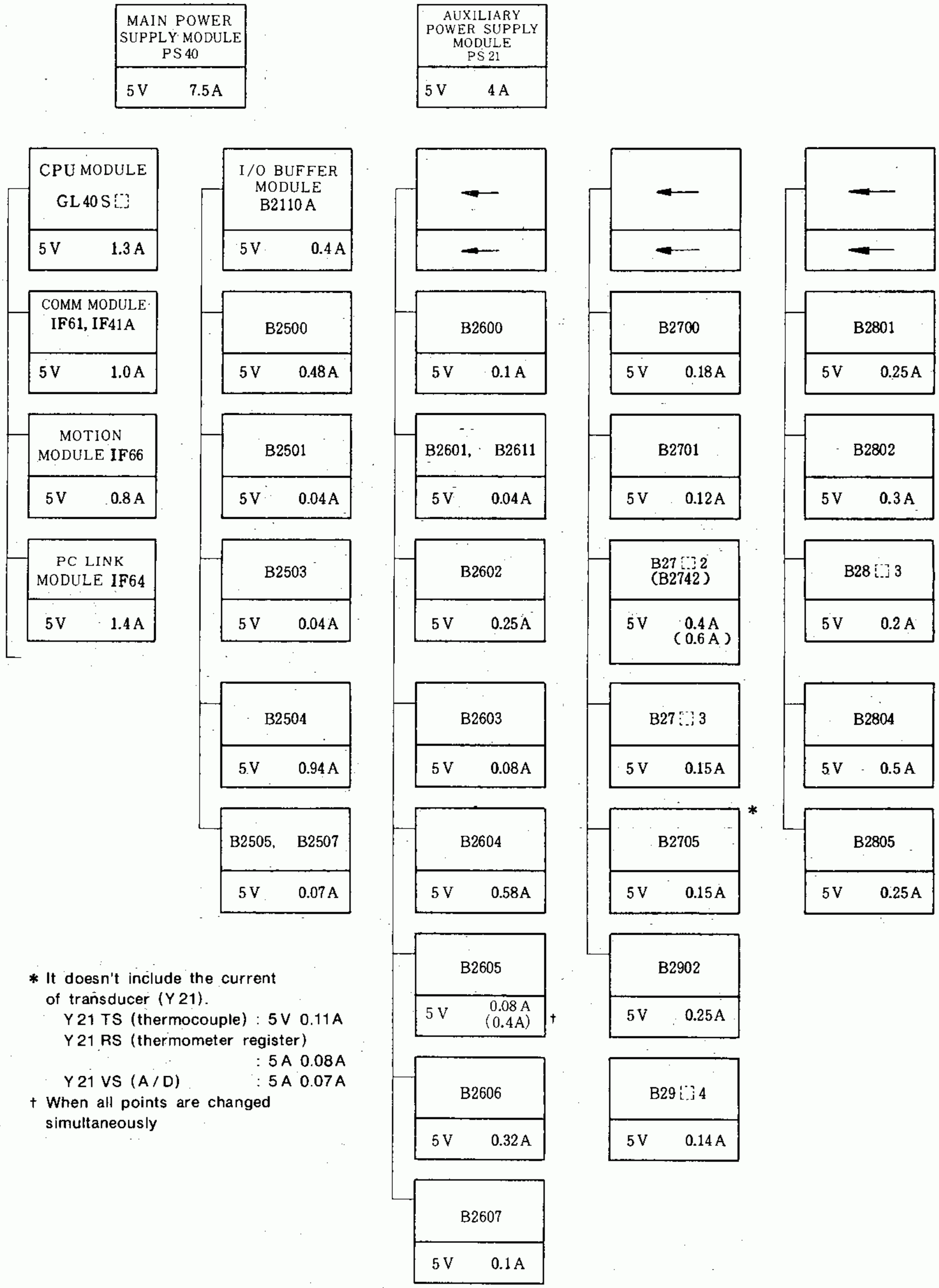
Fig. 8.20 External Power Supply for DC I/O Modules

If it is necessary to use a simple DC power supply such as full wave rectification, minimize the ripple by adding a smoothing capacitor. The following should be observed:

- Instantaneous output voltage with ripple should always be within the range of the operation voltage of the DC I/O modules.
- Output voltage, including that of the power ON time and power OFF time, should never exceed the transient voltage of the DC I/O modules.
- Prevent the introduction of surge voltage by adding a noise filter on the input side of a rectifying device.

8.3.5 Precautions when Installing I/O Module

The GL40S system can be provided with up to 8 I/O modules on rack 1 and up to 9 I/O modules on racks 2, 3, and 4. However, the number of modules to be installed must be limited so that the total consumed current of the modules to be used will not exceed the capacity of the internal power (supplied from a power supply module PS40 or a module PS21). The total consumed current should be calculated in accordance with Fig. 8.21.



* It doesn't include the current of transducer (Y21).
  Y21 TS (thermocouple) : 5V 0.11A
  Y21 RS (thermometer register) : 5A 0.08A
  Y21 VS (A/D) : 5A 0.07A

† When all points are changed simultaneously

Fig. 8.21 Consumed Current of Each Module

8.4 CONSTRUCTION, INSTALLATION AND WIRING OF CONTROL PANEL

The GL40S systems are delivered with the CPU module, the power supply module, the input and output modules, the mounting base, the cable, etc. separated from each other.

These components must be installed in a control panel housing, the construction, layout, and wiring of which shall conform to the following standards. Where more strict conditions for the wiring, etc. are specified by separate specifications, etc., there should be given priority. For further details not sepcified below, applicable status or regulations apply.

8.4.1 Construction of Control Panel

For the control panel, the following construction is recommended.

- Enclosed steel housing, self-standing (or wall-mounting)
- Dustproof, or semi-dustproof
- Cooling: Where the panel-interior temperature (GL40S ambient temperature) rises above 55℃, ceiling fan or other cooling devices must be used. A cooling fan should in principle be used to discharge air from the panel interior.

  The respective modules have the heating value given in Table 8.3, the heating value for I/O modules applies when all the 16 points are simultaneously ON.

- Dimension
  Determine the size, etc. of the control panel by referring to the dimension of the each unit modules (Appendix B) and the GL40S panel mounting dimension (Appendix C).

- Layout of mounting base
  Install these in the relative positions as shown in Appendix C, taking the cooling and other conditions into consideration.

Table 8.3 Heating Value and Weight of Module

Module (Type)	Heating Value W	Weight kg
CPU module (GL40S)	7	0.7
Main power supply module (PS40)	70	0.8
Auxiliary power supply module (PS21)	40	0.7
Servo I/F module (IF66)	4	0.6
COMM module (IF41A)	5	0.6
PCLINK module (IF64)	7	0.6
I/O buffer module (B2110A)	2	0.4
AC input module (B2501, B2503)	2	0.4
AC input module (B2505, B2507)	3	0.6
DC input module (B2601)	5	0.5
DC input module (B2611)	8	0.5
DC input module (B2603)	9	0.5
DC input module (B2605)	8	0.6
DC input module (B2607)	6	0.5
AC output module (B2500)	9	0.6
AC output module (B2504)	10	0.8
DC output module (B2600)	10	0.8
DC output module (B2602)	7	0.5
DC output module (B2604)	9	0.6
DC output module (B2606)	3	0.5
Numerical input module (B2701)	4	0.5
Numerical output module (B2700)	3	0.5
Analog input module (B2703, B2733, B2743)	6	0.5
Analog output module (B2702, B2712, B2722, B2732)	3	0.6
Analog output module (B2742)	5	0.6
Relay contact output module (B2902, B2904, B2914)	8	0.6
Instrumentation input module (B2705)	3	0.7
Reversible counter module (B2801)	3	0.7
Preset counter module (B2802)	5	0.6
Positioning module (B2803, B2813, B2823)	9	0.6
MEMOLINK master (B2804)	4	0.6
MEMOLINK slave (B2805)	2	0.6
I/F (B2806)	2	0.6
PID module (B2800)	5	0.6
Mounting base (MB40)	—	1.4
Mounting base (MB22)	—	1.3
I/O cable (W20-1)	—	0.5
I/O cable (W20-2)	—	0.3

8.4.2 Device Configuration in Control Panel

The modules of the GL40S are mounted on two types of mounting bases. Each module must be installed on any location of the mounting base.

When installing the GL40S in a control panel, determine the device configuration by considering the layout of other devices and the following. In Appendix C, a sample layout of the GL40S in a control panel is shown.

(1) Mounting Bases and I/O Cables

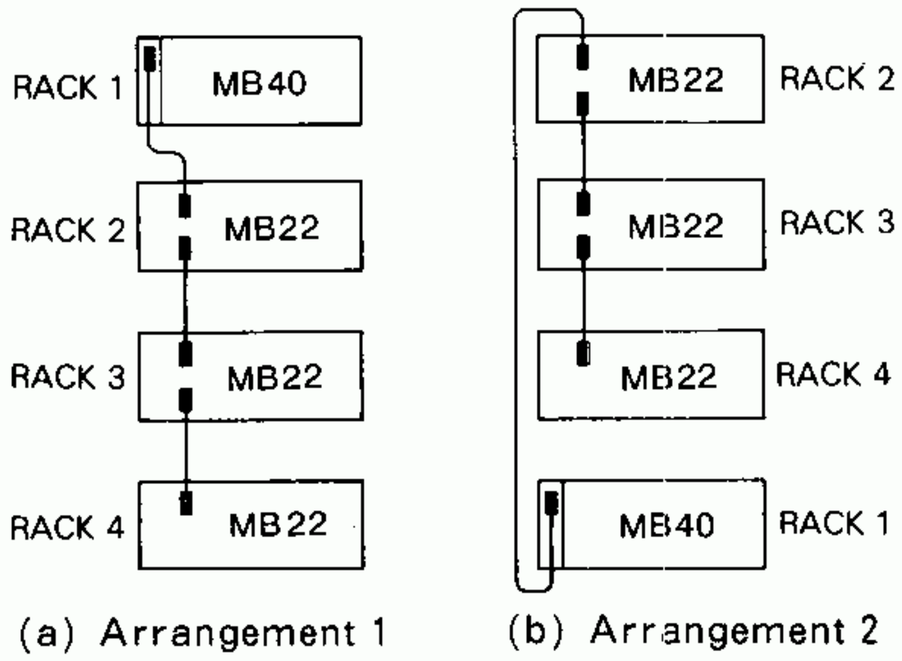


Fig. 8.22 Mounting Base Arrangement

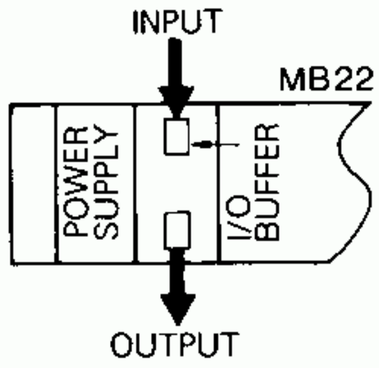
Mounting bases must be connected in the order shown in Fig. 8.22 : Racks 1 to 4. The I/O cables shown in Table 8.4 are used for communication between mounting bases.

Table 8.4 I/O Cable Specifications

Type JZMSZ-	Length m	Application
W20-1	0.5	Used for connecting across each mounting base (MB40, MB22), respectively.
W20-2	1.5	

* Three cables of type W20-2 can be used.

As for the two connectors provided on MB22 I/O buffer module, one on the top side is used for input lines and the other on the bottom side for output lines.



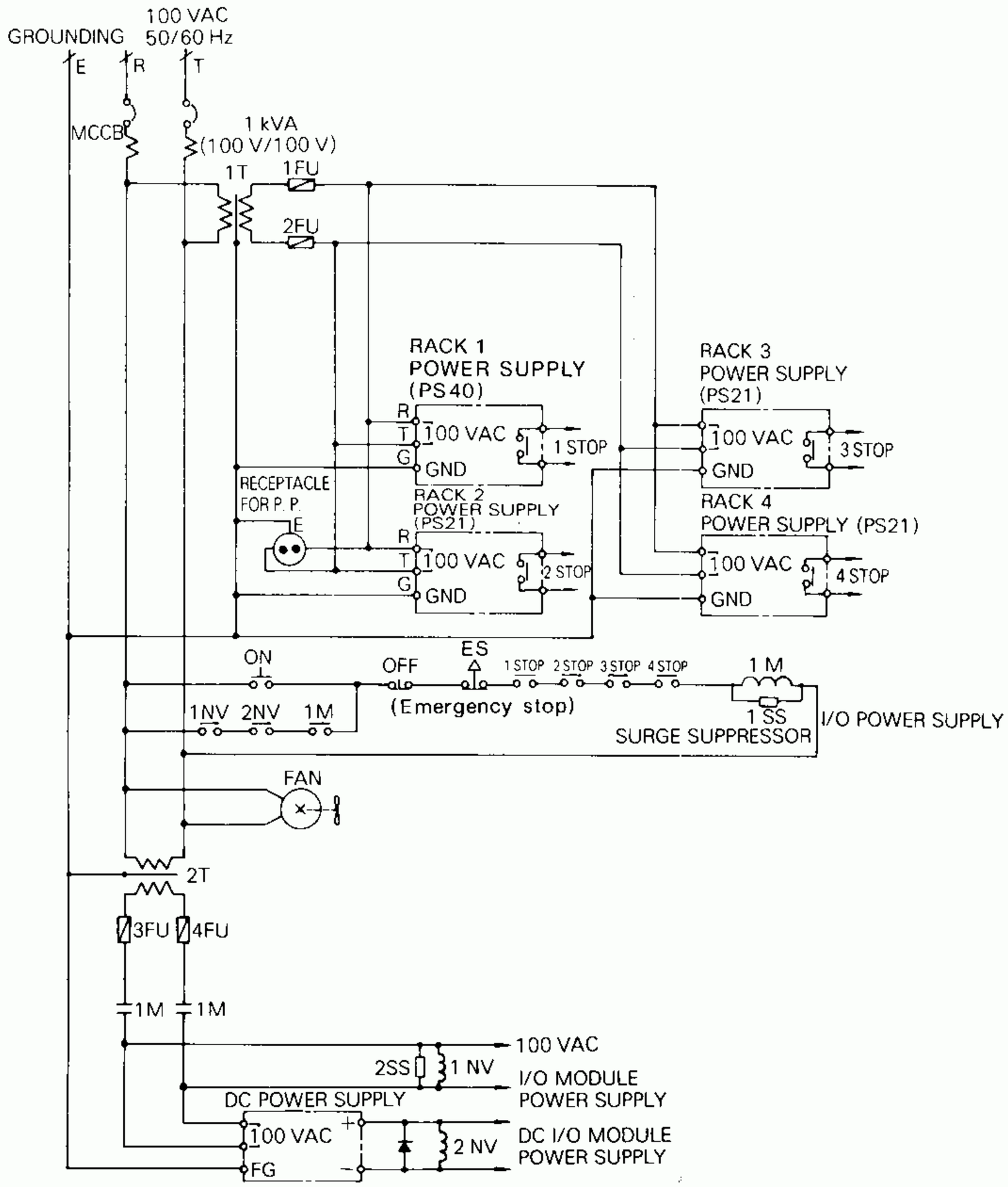
(2) Electrical Noise

- Avoid installing GL40S together with elements or wires carrying high-voltage and large current power* in the same panel.
- When installing GL40S together with low-voltage main circuit† in the same panel, install the elements and wires related to the low-voltage main circuit as far apart from the GL40S and its wiring as possible.
- Do not bind the GL40S wiring together with general control circuit‡ wiring.
- Install the mounting base to a solid steel panel (frame). Never install these on an insulator. When the panel (frame) is painted, remove the paint from the area around the mounting holes before installing the base, in order to secure good grounding, and to prevent noise.

* Above 600 VAC, 750 VDC or 800 A
† Below 600 VAC or 750 VDC, with a current above 20 A
‡ Below 600 VAC or 750 VDC, with a current below 20 A

(3) Power Supply Circuit

- Fig. 8.23 shows the example of power supply circuit.
- **When the power supply is in an unfavorable condition, connect a noise filter or an insulating transformer to the power supply line of a power supply module, I/O modules. Primary and secondary wirings of noise filter and transformer must be separated.**
- The voltage and capacity of the power supply depend on the types of I/O modules and the connected loads.
- **GL40S starts deciphering processes immediately when the power supply is turned ON. Before turning on the I/O module power supply, be sure that GL40S controls I/O without an error.**
- **The GL40S CPU is provided with a self-diagnosis function of stopping operation and turning output OFF when an error occurs. However, some faults and misoperations are not detected, and may destroy machines and devices. Set the interlock such as emergency stop externally to turn off the I/O module power supply when emergency occurs.**



Note
1. This example includes the minimum of necessary functions. Other required functions should be added according to each application.
2. Writing should be separated the primary side of a transformer from the secondary side.
3. "STOP" contact output from auxiliary power supply (PS40, PS21) is closed under normal I/O control of the GL40S.

Fig. 8.23 Example of Power Supply Circuit

(4) Wiring in Panel

The wiring related to GL40S in the panel is in types shown in Table 8.5. Use the wires of the listed sizes.

Table 8.5 Types of Wiring in Panel

Type of Wiring	Wire Size mm^2	Description
Power Supply	1.25	To be connected to the power supply terminal "100VAC" of power supply module, via circuit breaker, etc.
I/O Signal	0.3–1.25	To be connected to I/O signal lines and I/O module terminals (two 1.25 mm^2 wires can be connected to one terminal).
Grounding	1.25	Connection between the GND terminal of the power supply module and the control panel housing (ground).

8.4.3 Grounding Wire

- The GND terminal of the power supply module should be connected to the control panel housing, as shown in Fig. 8.24 at E, and connecting point E should be connected to a ground pole.
- The grounding wire between point E and the ground pole should be larger than 8 mm^2 in the cross-sectional area, and should be as short as possible.
- The grounding resistance should be 100Ω or less. (Ordinary buiding frames may be used. However, do not use a ground wire or ground pole in common with power lines, motors, etc.)

Note When metal ducts, metal tubes or wiring racks are used, ground them in accordance with the accepted technical standards.

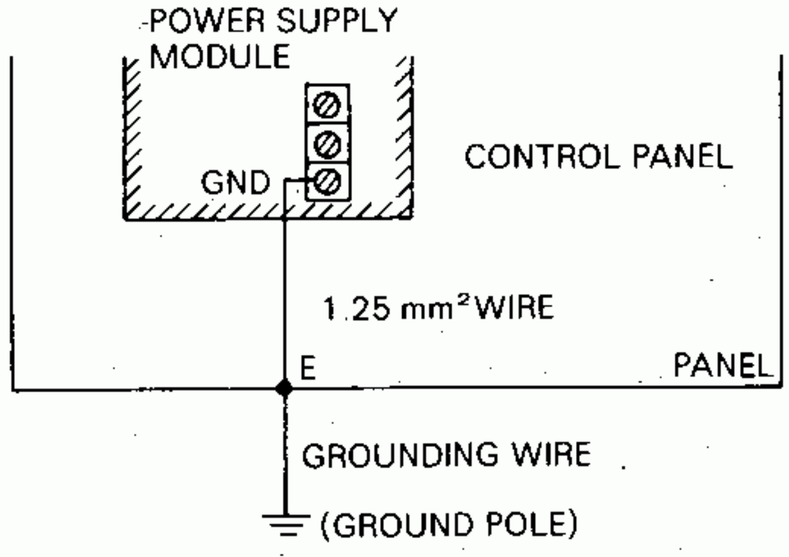


Fig. 8.24 Grounding Wire

8.4.4 External Wiring

(1) Cables for I/O Signal Lines

Cables to be used as external I/O signal lines should be selected in full consideration of the environmental conditions, mechanical strength, electrical noise, wiring length, operational voltage, etc. Isolate the I/O signal lines from each other and select cables on the basis of the guidelines given in Table 8.6.

(2) Installation of I/O Signal Line Cables

Since the I/O signal lines are low-voltage control circuit lines, separate these lines from ordinary control circuit lines and the main circuit lines as far as possible. Keep the space of minimum 10 centimeters between the lines. If they cannot separated, use the totally shielded cables and place the iron plate between the lines to separate them completely.

Table 8.6 I/O Signal Line Cable Installation

Wiring Distance	Description
30 m max	• DC output signal lines and DC input signal lines may be contained in the same duct. AC output signal lines and AC input signal lines also may be contained in the same duct. • DC I/O signal lines and AC I/O signal lines should be contained in their respective ducts, separately.
30 – 300 m	• DC input signal lines, DC output signal lines, AC input signal lines and AC output signal lines should be contained in their respective ducts, separately. • Where induced voltage is high, connect a dummy resistor or use the totally shielded cables. (Shield to be grounded at GL40S side.)
300 m min	• Do not use cables over 300 m, in view of the rush current to the output module. • Where the wiring distance is over 300 m, use a relay, and limit the wiring length between the realy and the control panel within 300 m.

8.5 SPARE PARTS

Generally, it is recommended that the following spare parts should be stocked.

- CPU module: one unit
- Main power supply module: one unit
- Auxiliary power supply module: one unit
- **Servo interface module: one unit**
- COMM module: one unit
- I/O buffer module: one unit
- I/O modules: At least one unit each of all the used modules. Where many modules are used, 3 % of the used number of modules are recommended as a spare parts quantity.

SECTION 9
GL40S MAINTENANCE

9.1 GL40S INSTALLATION PROCEDURE

The mounting bases, modules, and I/O cables are shipped separately. Follow the procedures described below when installing the GL40S in a control panel.

9.1.1 Installation of Mounting Bases

According to (1) of 8.4.2, determine the layout of mounting bases and drilled holes. Install wire ducts as necessary. There are two types of mounting bases. Install the mounting base by fastening four M5 screws through the four holes provided.

Note The mounting base connectors are covered. When installing the mounting base leave the cover installed over the connectors to protect them from foreign matter.

9.1.2 Installation of Modules

Install modules of the GL40S on the fixed mounting bases. Fig. 9.1 shows how to install a module on a mounting base.

Remove the connector cover. Fit the guide posts of the module into the guide holes of the mounting base and push the module in. Then fasten the module to the mounting base with the M4 screws provided with the module.

Note Do not remove the connector cover if no module is to be installed.

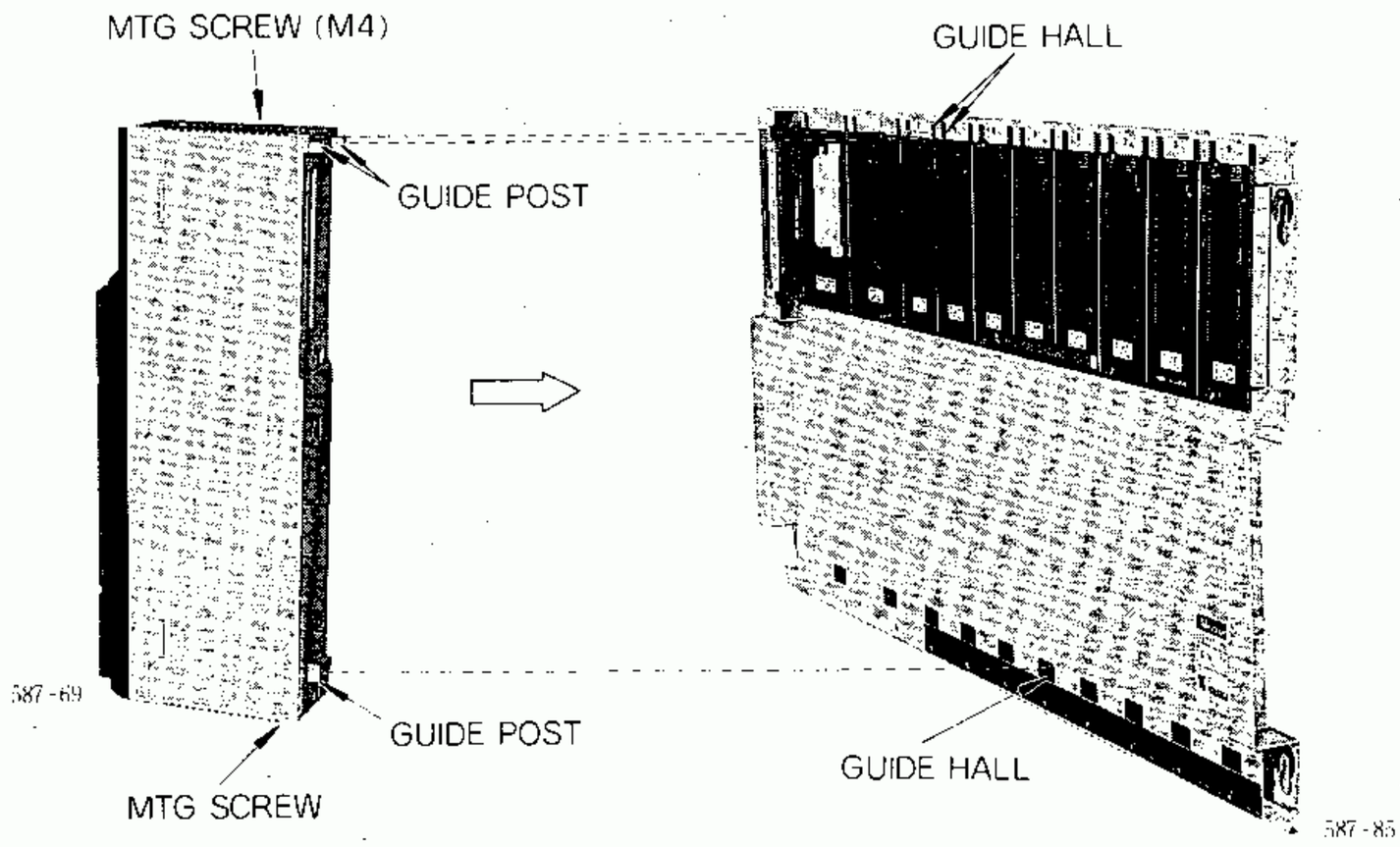


Fig. 9.1 Module Installation

The type of mounting base and the mounting location are determined depending on module types. Figs. 9.2 to 9.3 show mounting place of each module on the mounting base.

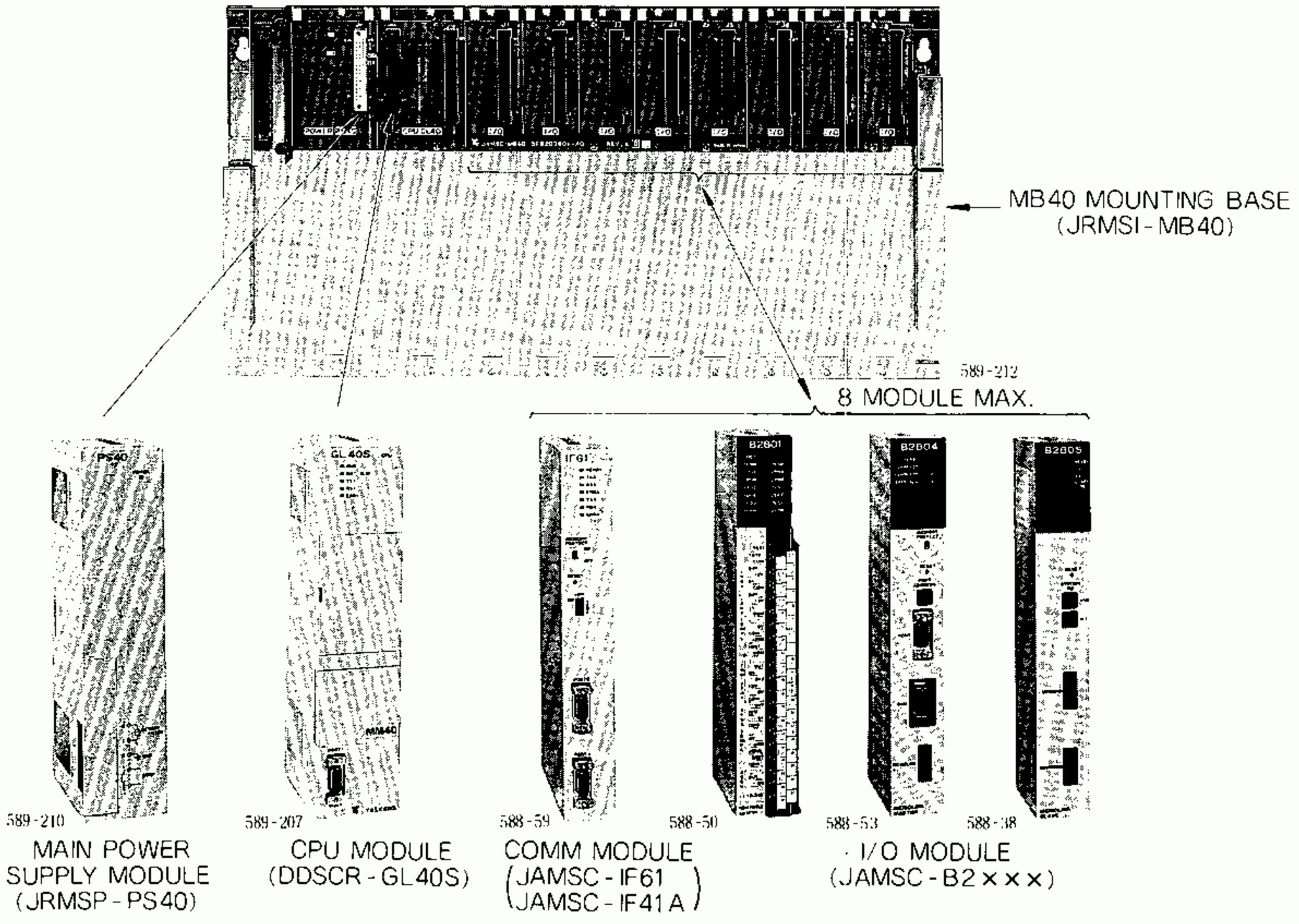


Fig. 9.2 Module Mounting on Mounting Base MB40

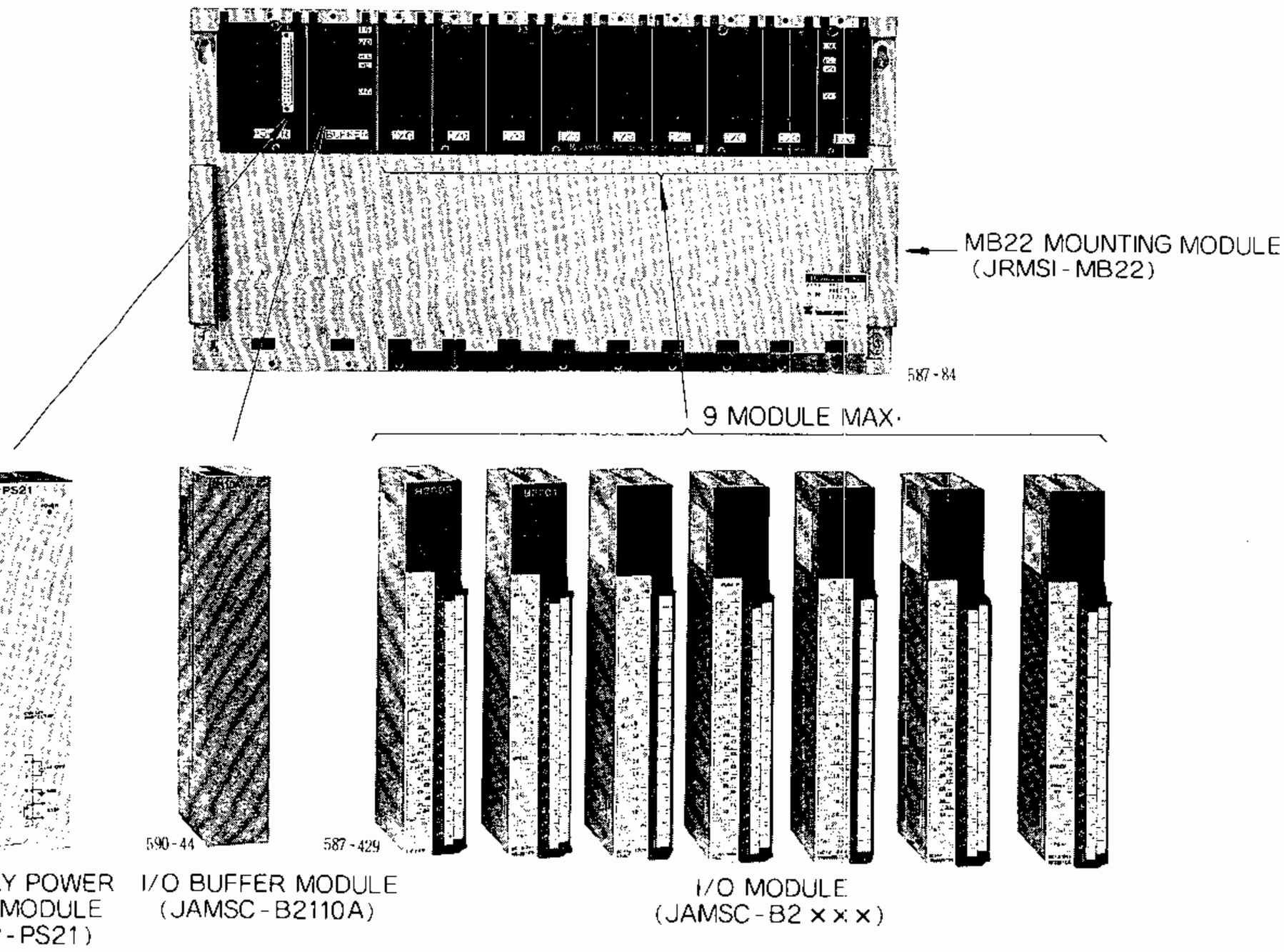
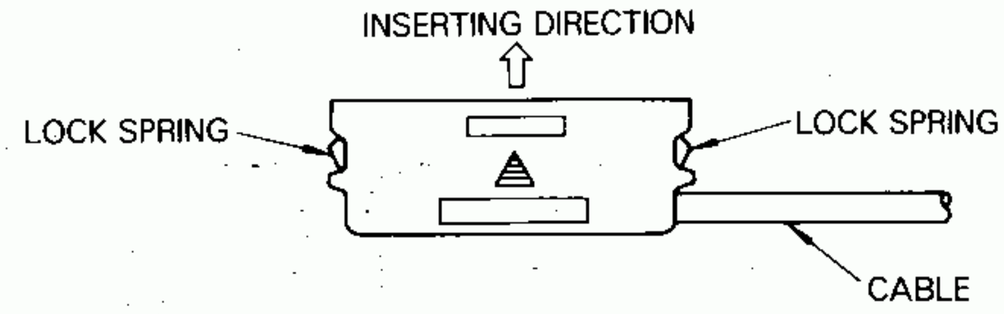


Fig. 9.3 Module Mounting on Mounting Base MB22

9.1.3 Connection of I/O Cables

Next, connect the mounting bases with I/O cables. Two I/O cables are provided: W20-1 (0.5m) and W20-2 (1.5m). Choose one according to setting conditions. As an option, W20-3 (3m), W20-4 (4m), W20-5 (5m) and W20-6 (6m) are available. Both ends of I/O cables have 50-pin connectors as shown in Fig. 9.4.

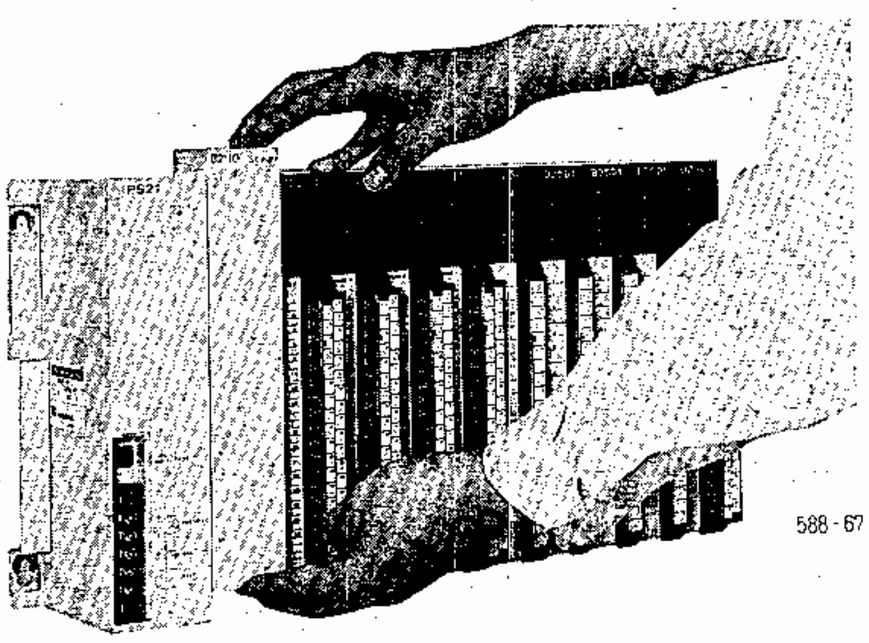


Note
1. I/O cable connector must be inserted until it is locked firmly.
2. Remove I/O cable, depressing both sides of connector lock spring.
3. Full length of W20 cable must be 6m max.

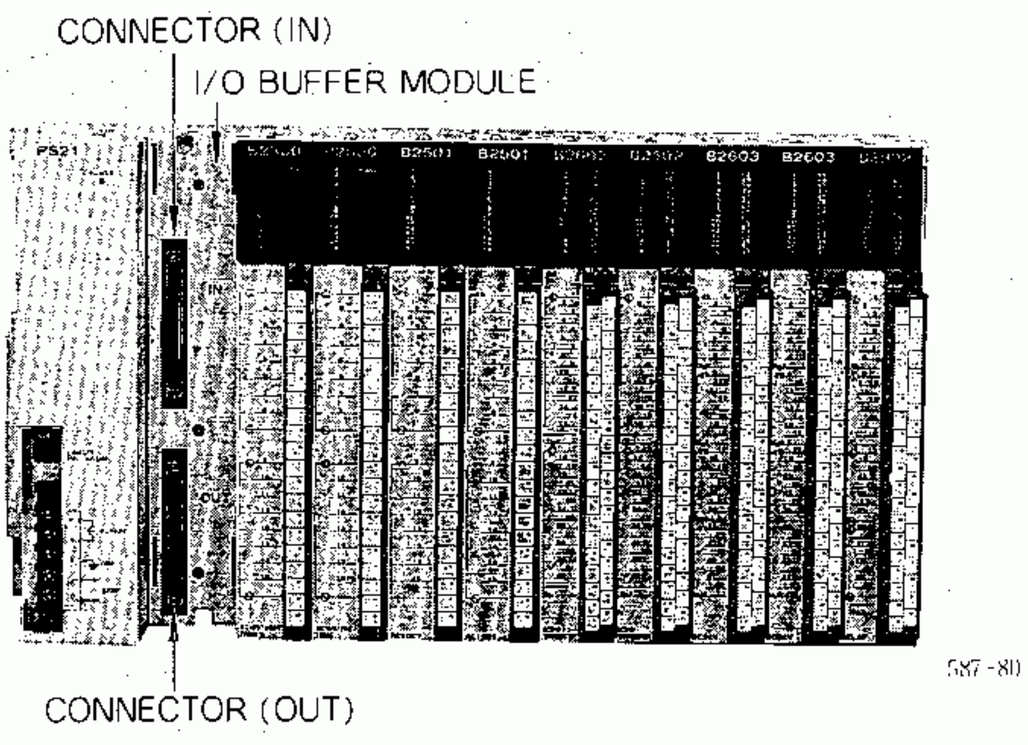
Fig. 9.4 I/O Cable Connector

For I/O cable connection, connector on the left of MB40 mounting base and connector on the front of I/O buffer module mounted on MB22 mounting base are used. (MB22 mounting base does not have its own I/O cable connector.)

Two connectors (IN and OUT) can be seen after removing I/O buffer module front cover.



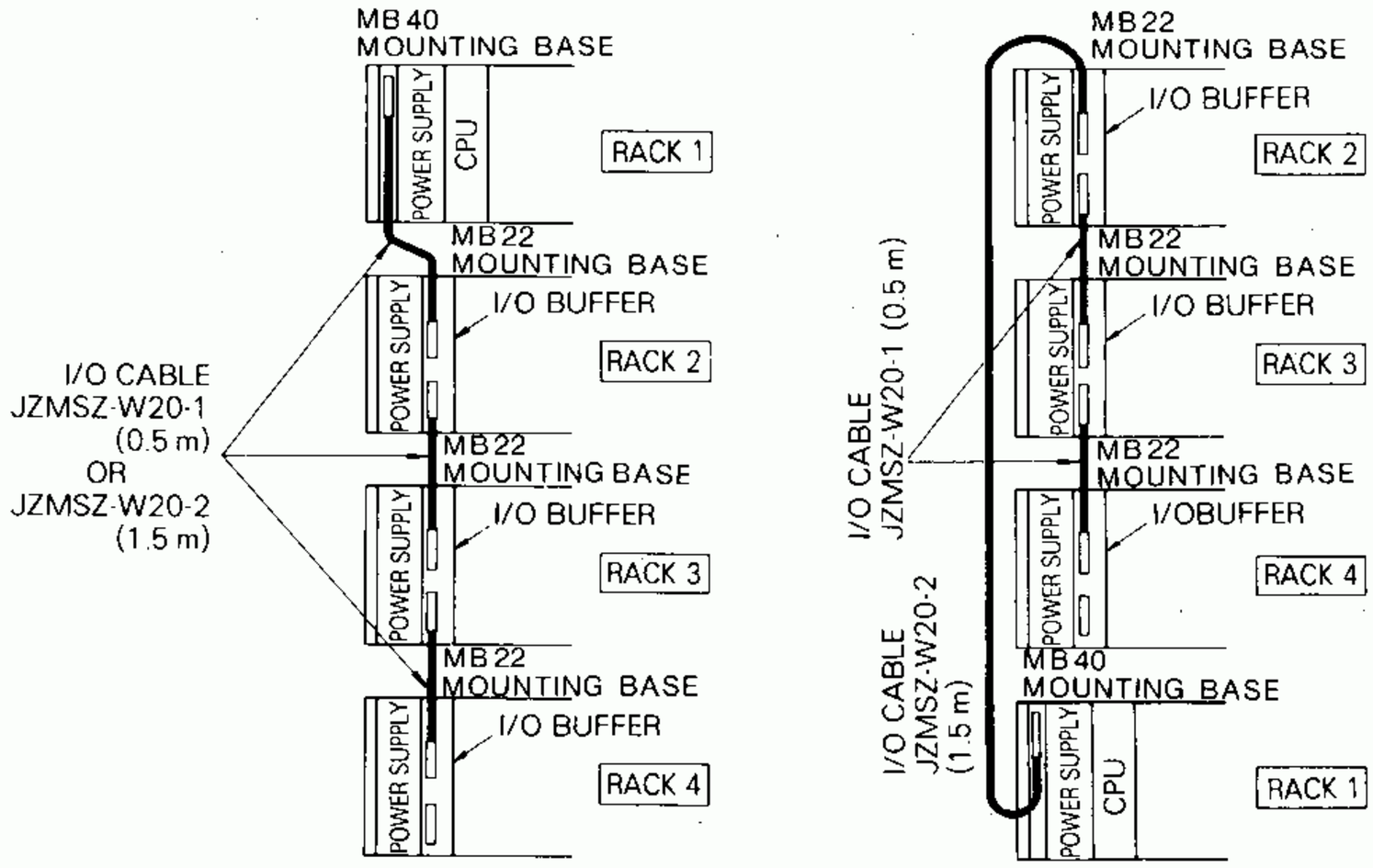
(a) Before Removing Front Cover



(b) After Removing

Fig. 9.5 I/O Buffer Module I/O Cable Connector

Connect the cables in the correct direction so that signal flows from output to input. Fig. 9.6 shows connection of I/O cables.



(a) Rack 1 at thr Top

(b) Rack 1 at the Bottom

Note
1. I/O cable connector must be inserted uniti it is locked firmly.
2. After completing I/O cable connection, provide covers to I/O buffer modules.

Fig. 9.6 I/O Cable Connection

9.1.4 Wiring of Modules

The power supply module and I/O modules have terminal blocks for connecting the power lines and I/O signal lines (Table 9.1).

Table 9.1 Terminal Blocks and Connectors for Wiring

No. of Pins	Power Supply Size	Applicable Pressure Terminals	Applicable Module Type	Remarks
20	1.25mm^2	Pressure terminal • R type: R1.25 – 3.5 • Fork type: E1.25 – 3.5	B2500, B2501	Removable from module
38			B2602, B2603 B2902	
5		Pressure terminal • R type: R1.25 – 4 • Fork type: 2 – 4Y F1.25 – 4	PS40, PS21	Not removable from module

Perform wiring to terminal block in the status with the terminal cover removed as shown in Figs. 9.7 and 9.8.

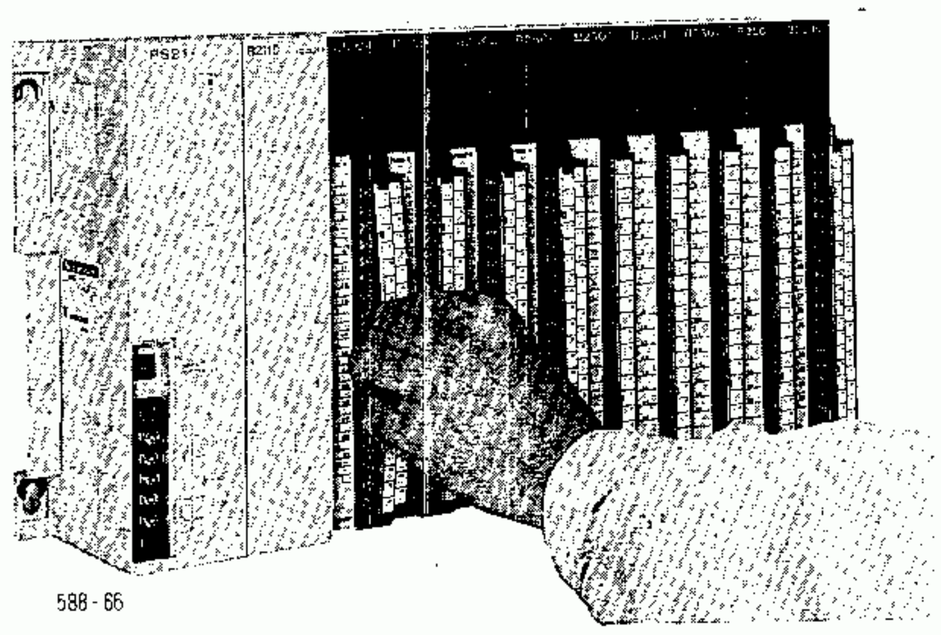


Fig. 9.7 Terminal Cover Removed

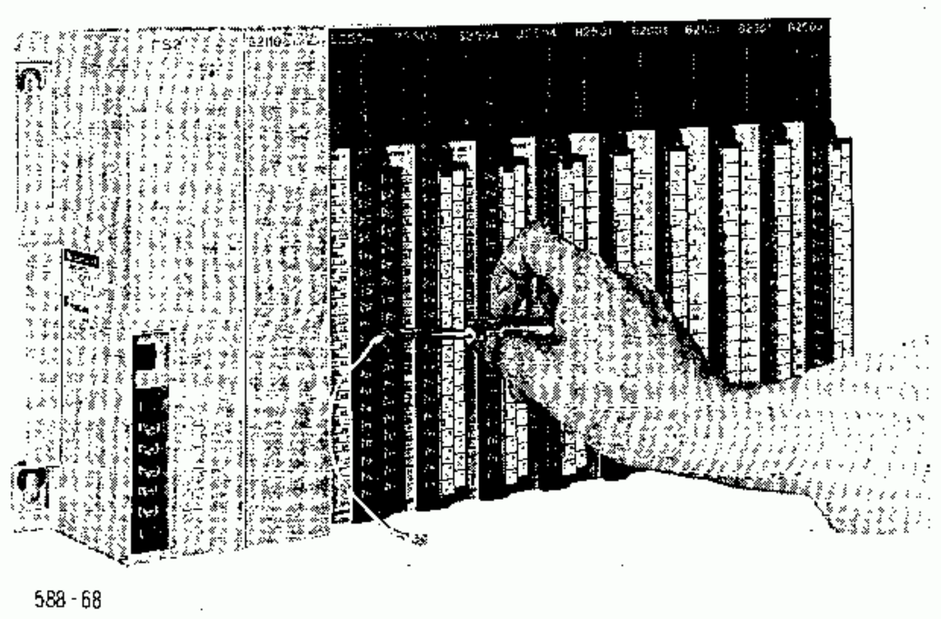


Fig. 9.8 Wiring to Terminal Block

9.2 COMM SETTING AND ERROR INDICATION

(1) When installing the COMM module, set the following modes by operating the DIP switches on the front panels of the modules.

① 1SW-2, 3 (Setting of mode during communication command execution)

Setting is needed when executing the communication command (COM). Set as follows depending on whether the transparent or MEMOBUS mode transmission is used during communication command execution.

Table 9.2 Setting of Transmission Mode during Communication Command Execution (1SW-2)

1SW-2	Setting
ON	Set Port 3 to transparent mode.
OFF	Set Port 3 to MEMOBUS mode.

Table 9.3 Setting of Transmission Mode during Communication Command Execution (1SW-3)

1SW-3	Setting
ON	Set Port 4 to transparent mode.
OFF	Set Port 4 to MEMOBUS mode.

② 1SW-4 (Setting of operation mode)

Set to the self diagnosis or normal operation mode. Must be normally set to "OFF."

Table 9.4 Operation Mode Setting

1SW-4	Setting
ON	Self diagnosis mode
OFF	Normal operaiton mode

③ 1SW-4

Must be set to "OFF".

(2) Error Indication

Status errors that occur inside the COMM module is indicated by the LEDs on the front panel of the modules.

Table 9.5 Errors and LED Indications

Errors	LED Inidcation			Remarks
	READY	ERR1, 3	ERR2, 4	
ROM Error	◑	◑	○	○: Lit ●: Extinguished ◑: Blinked
RAM Error	◑	○	◑	
Common Memory Error	◑	◑	●	
Watchdog Error	◑	◑	◑	

9.3 MODULE REPLACEMENT

The indicator lamps of the modules and the programming panel permit locating a defective module.

It is possible to locate and replace the defective module so quickly that system down time will be minimized.

Follow the procedures given below to replace a module.

(1) Turn off supply power.

Remove the AC power from the GL40S and external supply power from the I/O modules. If I/O modules are replaced with the power ON, outputs might be erroneous.

(2) Remove the terminal block and connector. [Fig. 9.9 (a)

It is not necessary to remove the wires separately. Rather, remove the terminal block. Remove the terminal block two screws, top and bottom.

Remove the power lines one by one since power supply module terminal block is not removable.

(3) Loosen the screws fastening the module. [Fig. 9.9 (b)

Loosen the two screws, top and bottom, fastening the module to the mounting base, until they are released from the base.

(4) Remove the module. [Fig. 9.9 (c)]

Holding the top and bottom of the module, pull it out toward you.

(5) Install a new module.

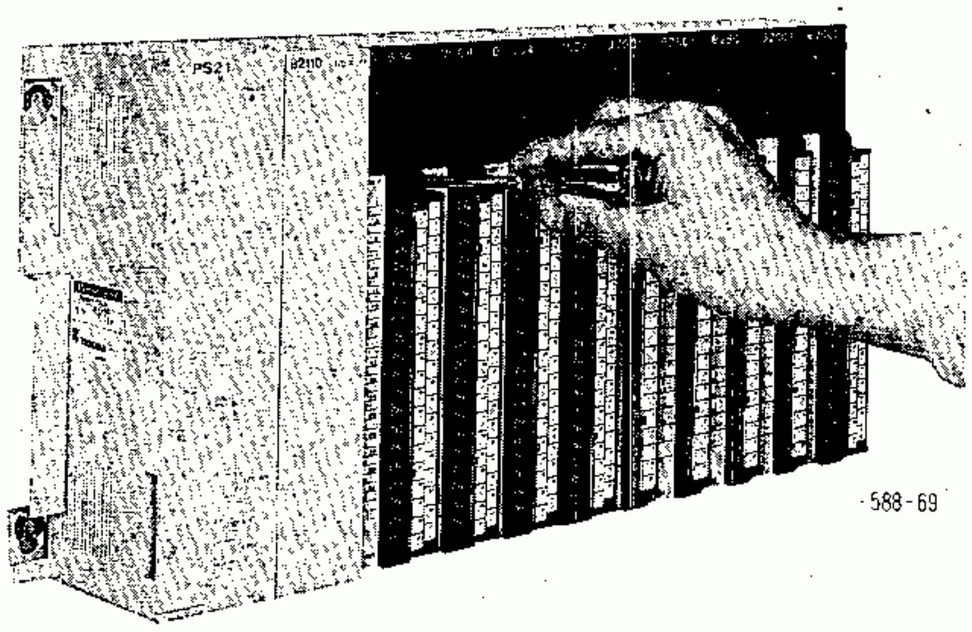
Fit the guide posts of the module into the guide holes of the mounting base and push the module in so that the guide posts slip into the holes completely. Then fasten the module to the mounting base with the screws provided with the module.

(6) Install the terminal block and connector.

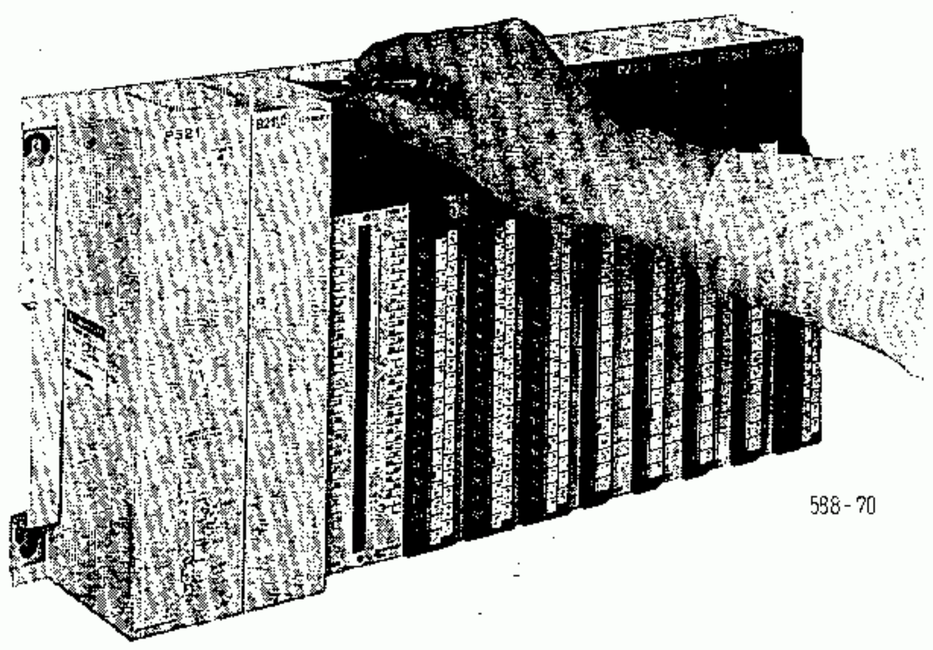
Replace the terminal block removed in step (2).

(7) Turn on power.

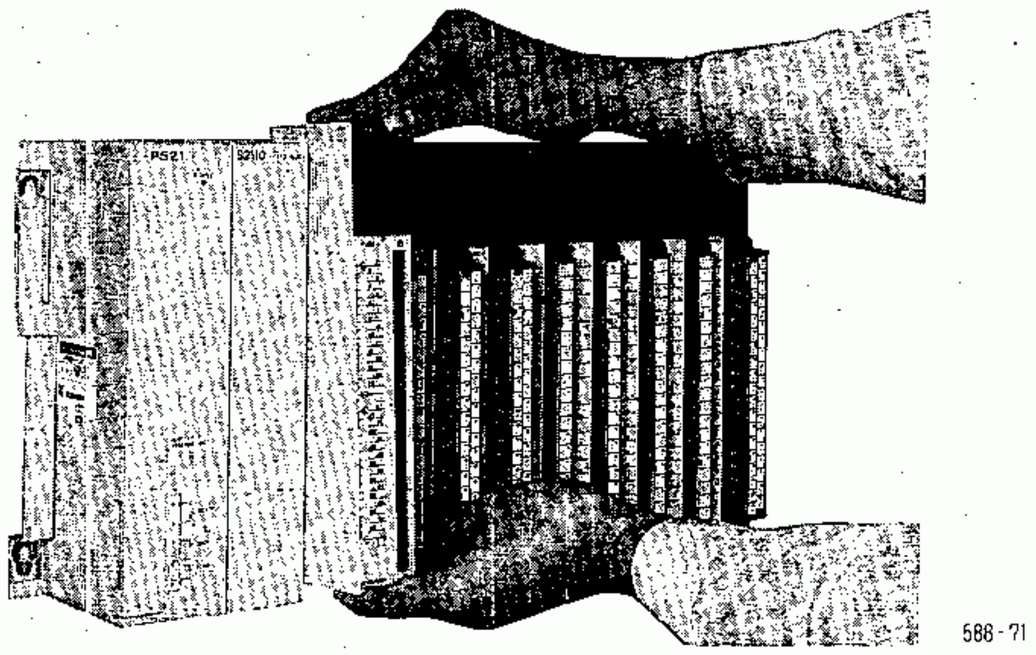
Make sure that the module and wiring are correct, before turning on power.



(a) Removing Connector Terminal Block



(b) Loosening Module Mounting Screws



(c) Pulling out Module

Fig. 9.9 Module Replacement

9.4 BATTERY REPLACEMENT

Back-up power for the GL40S memory is provided by battery. It is recommended to replace the battery every two years.

Note Battery service life varies with the environmental conditions (temperature and humidity) and working time (AC power failure time).

(1) Battery Specifications

Table 9.6 gives the specifications of the battery used in the GL40S.

Table 9.6 Battery Specifications

Item	Specifications
Name	Lithium battery
Type	BR-2/3A-1
Manufacturer	Matsushita Battery Industry Co., Ltd.
Nominal Voltage	3V
Nominal Capacity	1200 m Ah
Ambient Temperature	0 ℃ to + 55 ℃
Storage Temperature	− 20 ℃ to + 45 ℃
Life	• Warranty: 5 years at 25 ℃ • Actual protective period for memory: 1 year at 25 ℃
Approx Weight	15g

Note When the battery in the Table above is required, contact Yaskawa representative.

(2) Battery Replacement Interval

The battery will last for a maximum of five years or until the total time of AC power failure reaches a year. If the "BATT ALARM" lamp of the CPU module lights, replace the battery within one month.

Approximate standards for battery replacement are given below.

Depending on GL40S average daily operating hours.

- 12 hours/day: replace every two years.
- 16 hours/day: replace every three years.
- 20 hours/day: replace every four years.

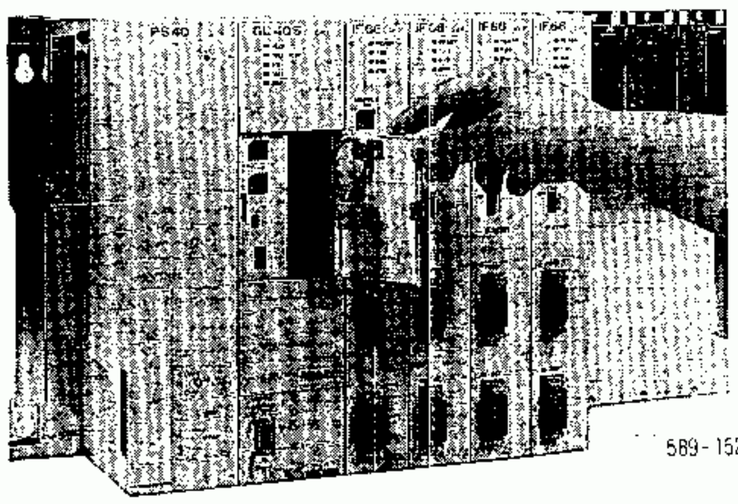
(3) Battery Replacement Procedures

Be sure to replace the battery with the CPU module connected to the mounting base and receiving AC power to the main power supply module. If the battery is removed with no AC power supplied, the memory contents will be destroyed. But if the battery is removed with no AC power supplied for a short time (within 30 minutes), the memory contents will not be destroyed.

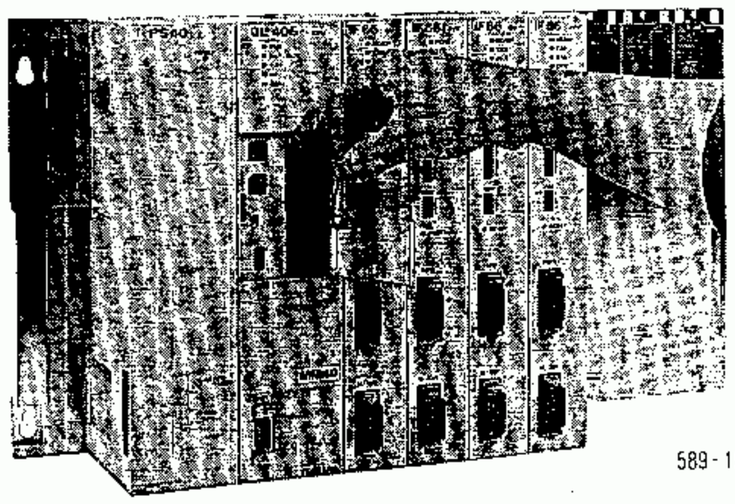
Replace the battery using the following procedures:

① Preheat the soldering iron.
② Confirm that the POWER lamp of the main power supply module is lit (AC power ON).
③ Remove the front cover. See Fig. 9.10 (a).
④ Take out the battery from the battery holder (Fig. 9.10 (b)), then separate the connector attached at the ends of the leads from the CPU module. See Fig. 9.10 (c).
⑤ Remove the leads from the battery, using the preheated soldering iron.
⑥ Solder the removed leads to the new battery, using care to observe the polarity. (Lead colors Red: Positive; Black: Negative)
⑦ Place the new battery in the battery holder and insert the connector attached at the ends of the leads into the CPU module connector.
⑧ Confirm that the "BATT ALARM" lamp of the CPU module goes off.
⑨ Provide the battery cover and turn off the power supply for the soldering iron.

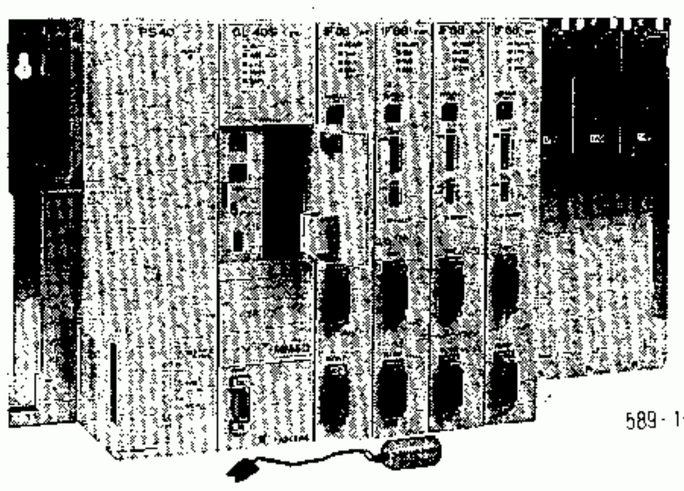
This completes the battery replacement.



(a) Opening CPU Module Front Cover



(b) Taking out Battery from Battery Holder



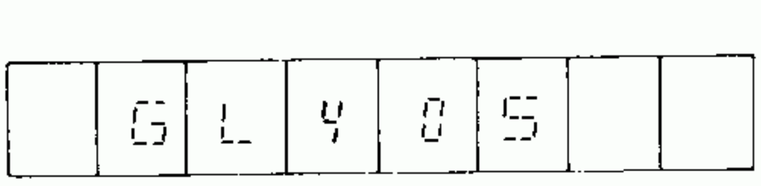
(c) Separating Connector from CPU Module

Fig. 9.10 Battery Replacement

9.5 REGISTER ACCESS PANEL

Connected to the connector on the front panel of the COMM module, the register access panal (RAP) is used to display ON/OFF states of coils and input relays, simulated operation (forced ON and OFF) and register data and to set and change data to holding registers.

Fig. 9.11 shows the register access panel. The RAP displays as follows when power is turned on.



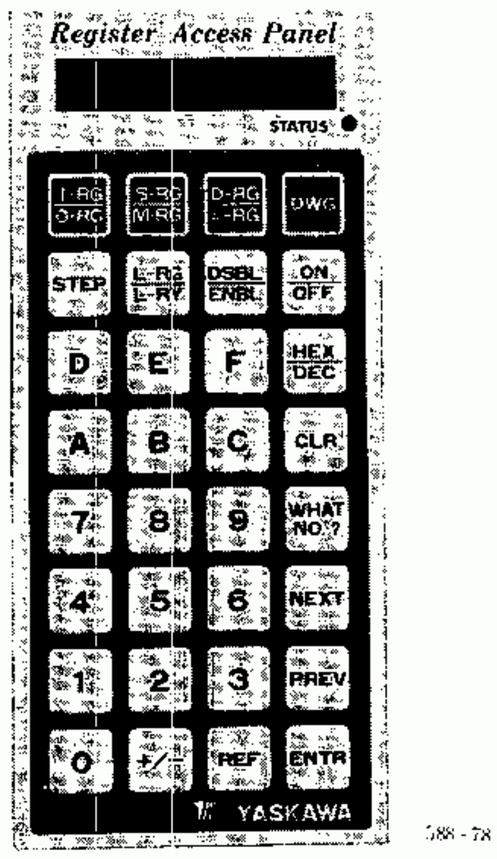


Fig. 9.11 Register Access Panel

9.5.1 Operation Keys

Table 9.7 RAP Operation Keys

Key	Function
0 to 9	Numerical keys to input Nos. and numerals. Keys A to F are invalid in decimal input.
A to F	
+/−	① Set whether to monitor registers with or without signs. Valid only for decimal monitoring. (Invalid during monitoring of step timers and port parameters.) ②Set whether to input values in registers without signs or as positive and negative numerals. Valid for decimal input. (Invalid during step timer and port parameter changes.)
CLR	Depress this key to clear display to "−". Depressed to reset error display or to cancel key function during operation.
DSBL/ENBL	Used in simulated operation of coils and input relays. Depress this key to set up simulated state (DISABLE). By depressing it again, the simulated state is released (ENABLE). The state reverts each time the key is depressed. Valid if the memory protect switch is OFF. Steps cannot be changed forcibly.

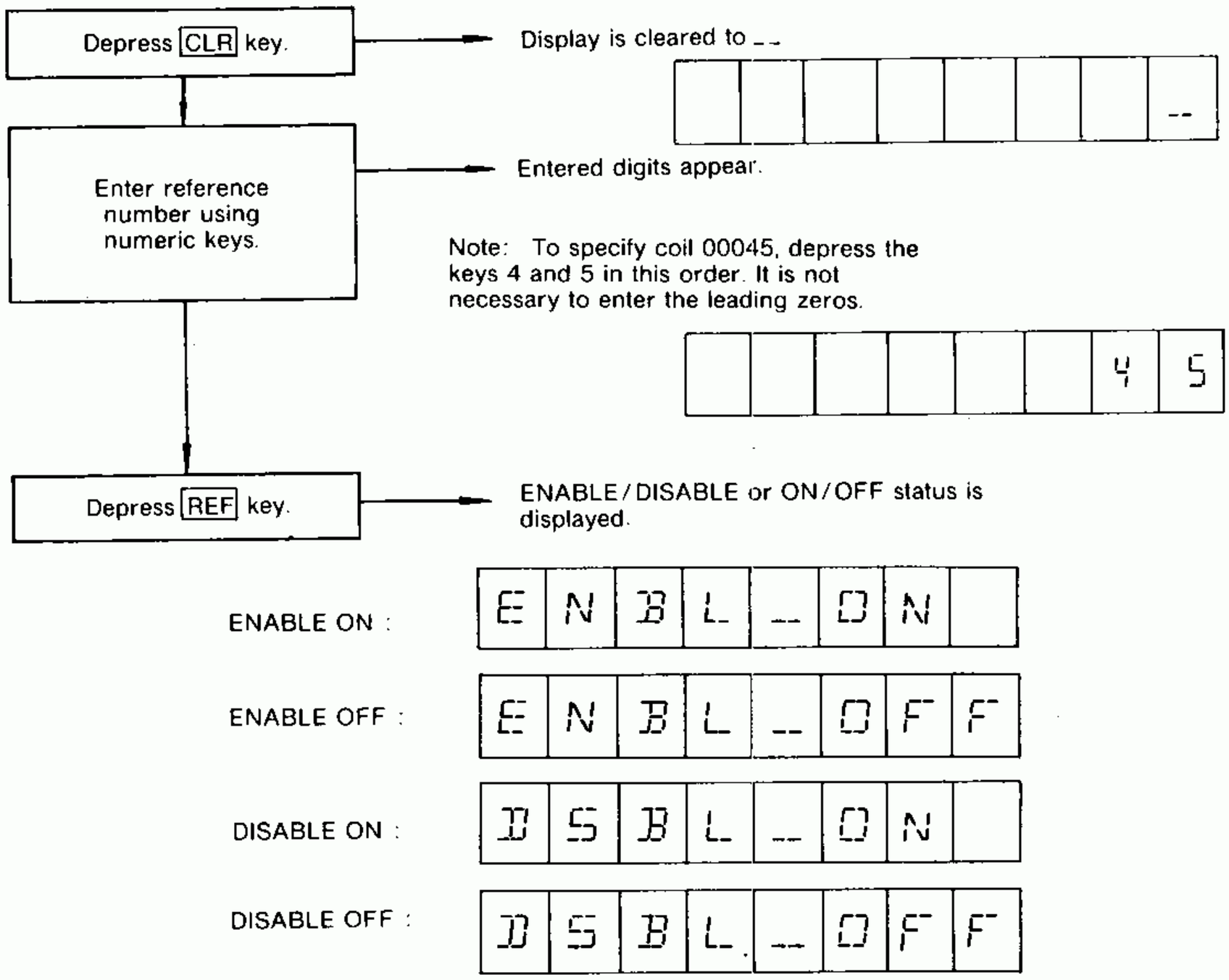
Table 9.7 RAP Operation Keys (Cont'd)

Key	Function
ON / OFF	Forcibly switches coil or input relay state in a simulated condition on or off. The state reverts each time the key is depressed. Valid if the memory protect switch is OFF. Steps cannot be forcibly switched ON and OFF.
HEX / DEC	Changes type of values and indicated value input (decimal or hexadecimal). Depress this key to alternately change from decimal to hexadecimal and vice versa.
STEP	The key sets step reference. Depress this key to clear indication and function, indicating only STEP and the cursor. Depress again to leave only the cursor.
ENTR	This key stores input value in the holding register or sets as a port parameter. Valid if memory protect switched off.
WHAT NO?	Depressed to show which reference No. or address is monitored at present.
NEXT	Depress this key to monitor data of reference No. or address next to what is being monitored at present.
PREV	Depress this key to monitor data of reference No. or address prior to what is being monitored at present.
REF	Depress this key to monitor data of set reference No. and address.
L-RG / L-RY	This key sets references for link registers and link relays. Depress once to set up link register reference acknowledge state; depress twice to set up link relay reference acknowledge state; and depress three times to normal reference acknowledge state.
I-RG / O-RG	
S-RG / M-RG	Key are always invalid with GL40S.
D-RG / #-RG	
DWG	

9.5.2 Operation

(1) Displaying ON/OFF Status of Input Relay 1 ×××× or Coil 0 ××××

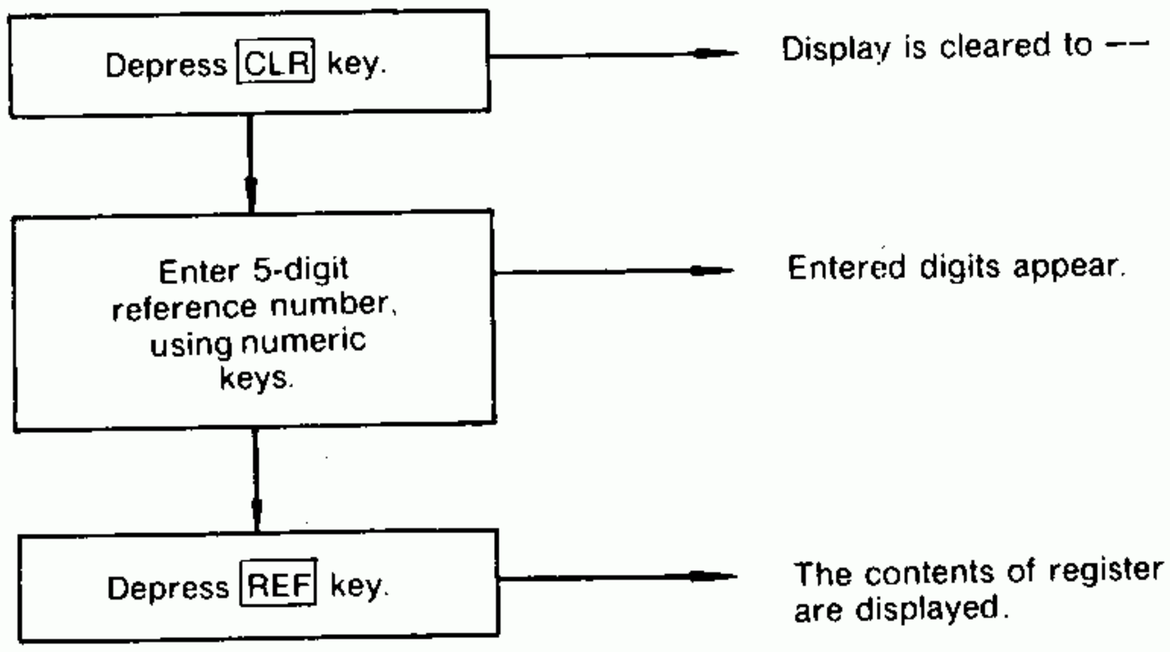
Set reference number, and the status for example, of the specified coil will be displayed enabled (observing ladder logic) or disabled (turned on or off independently of ladder logic).



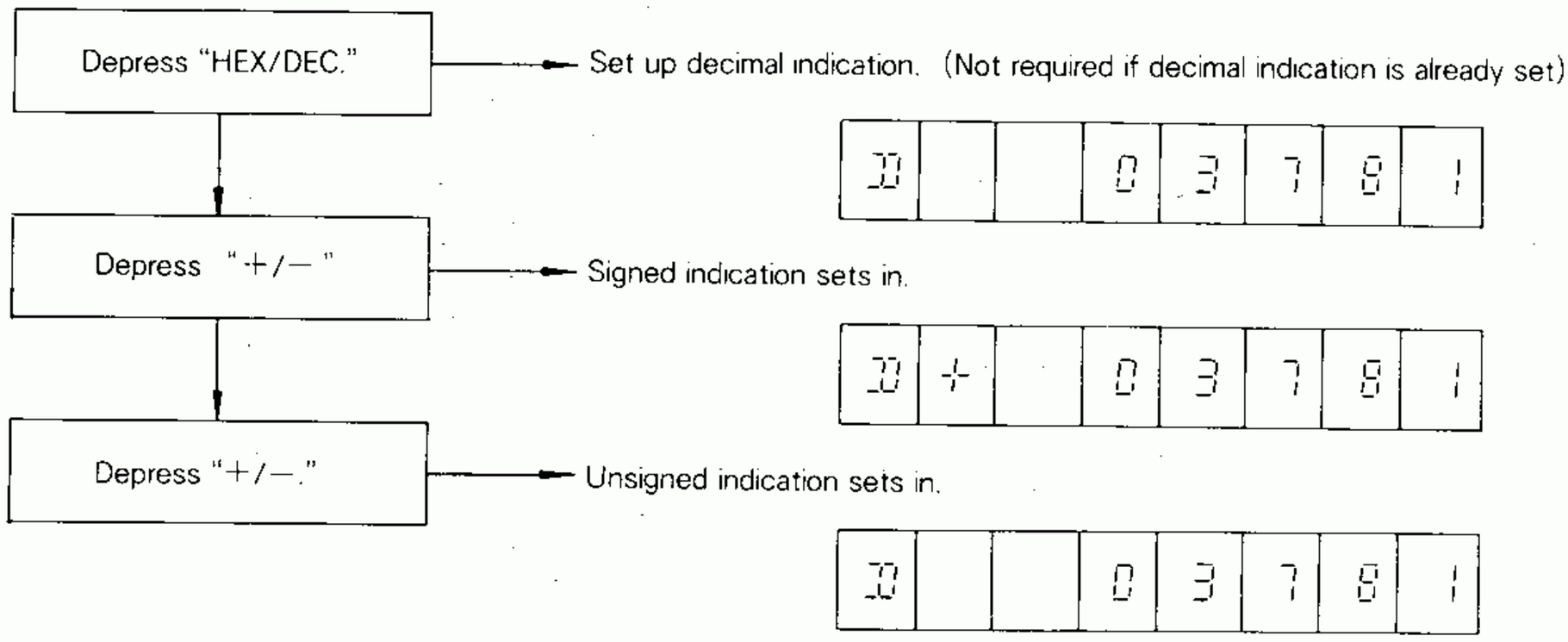
- The status display is refreshed in every scanning cycle.
- If you push NEXT (PREV) at this time, the reference number is increased (decreased) by one and the status of the input relay or coil identified by the updated reference number appears.

(2) Displaying Contents of Input Register 30 ×××, Constant Register 3 ×××× or Holding Register 4 ××××

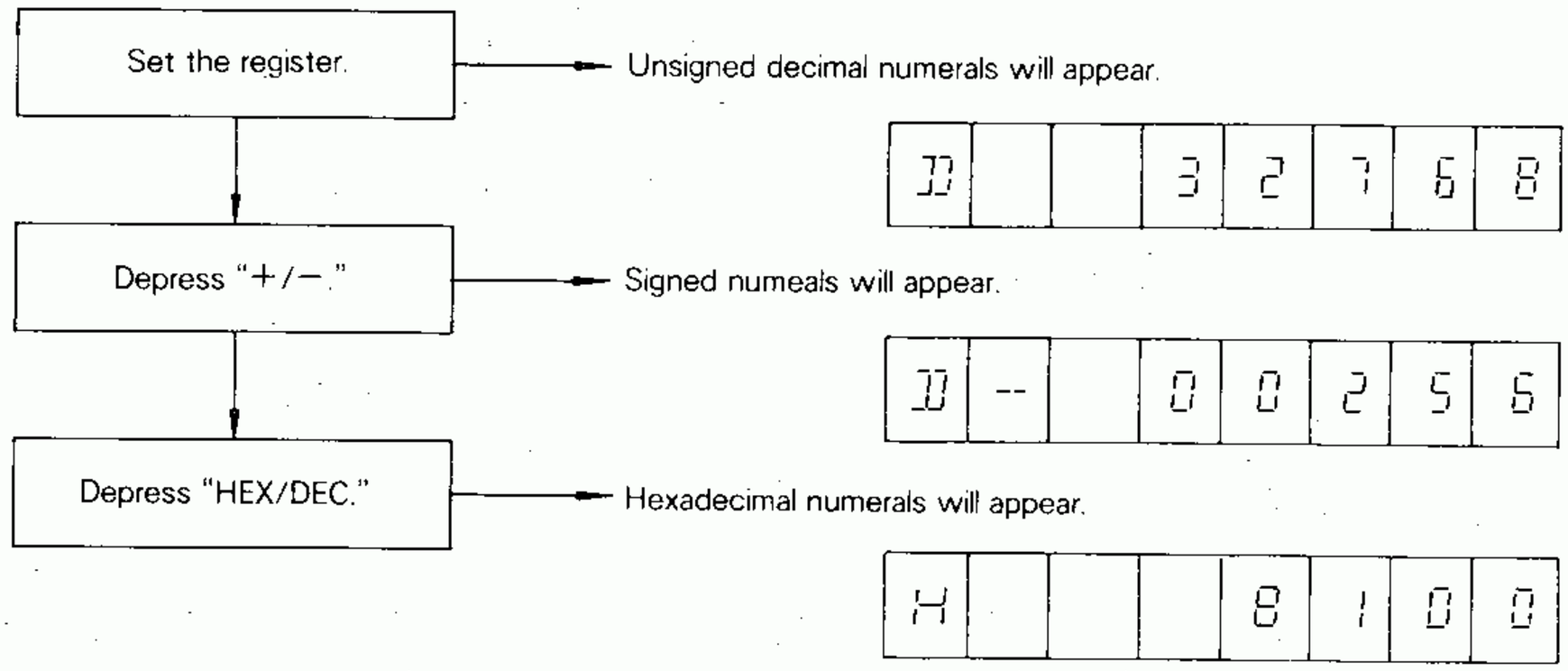
Set a reference number, and the contents of the register will be displayed. It is possible to select decimal or hex for the displayed data.



- The data will be refreshed in every scanning cycle.
- The reference number is increased (decreased) by one every time you push [NEXT] ([PREV]).
- Display of the data changed over in decimal form and hex form alternately as you push [HEX/DEC]. The decimal mode is selected first.
- Signed indication is operated as follows.



- By indicating negative numerals hexadecimally, "1" sets in the most significant bit. If negative numerals are indicated without sign, hexadecimal numerals are converted into decimal numerals as is and unusual numerals will be indicated. An example of register data of -256 (256 is 100h hexadecimally) is shown below.

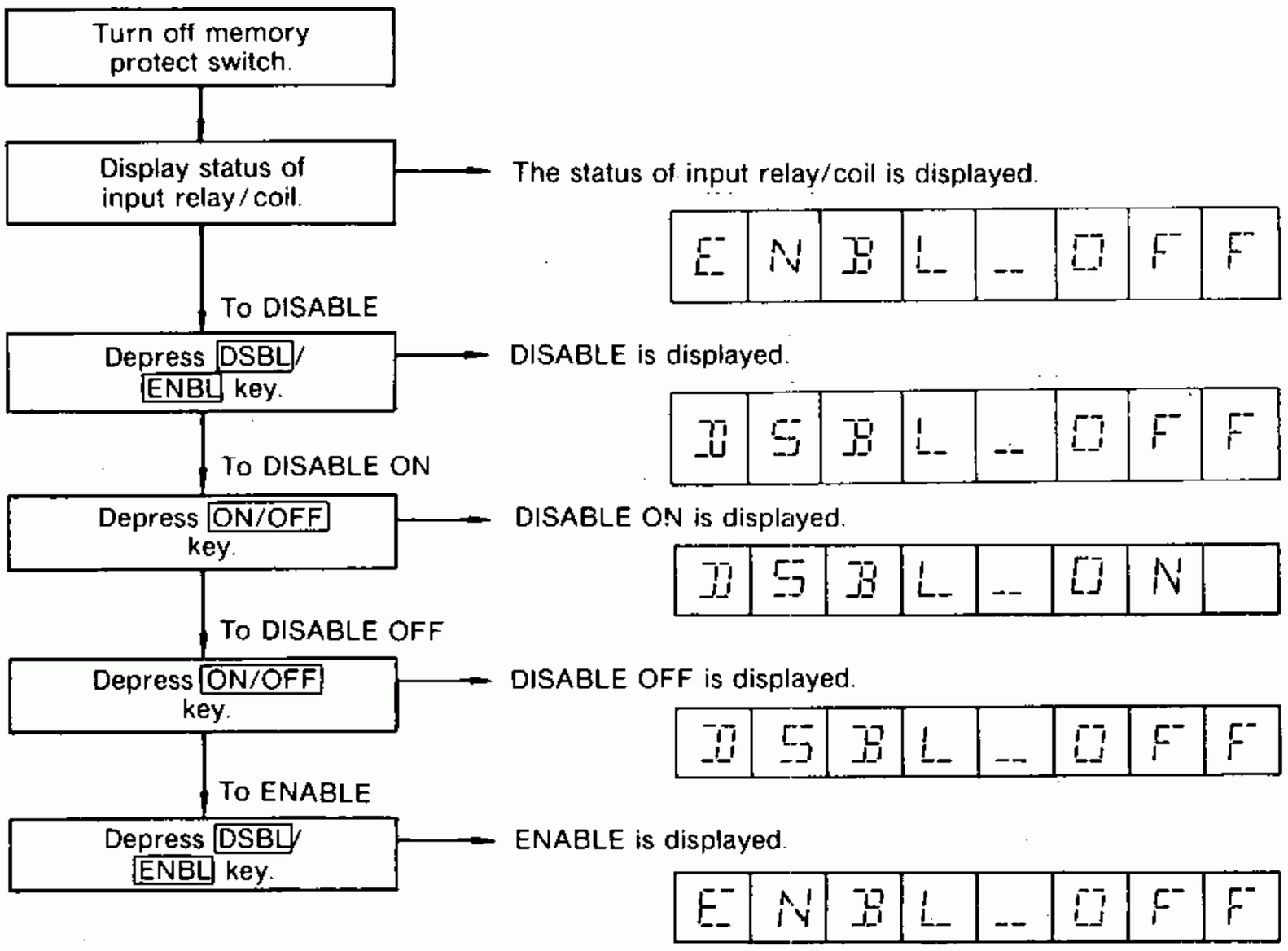


(3) DISABL/ENABL of input Relay or Coil

It is possible to DISABLE or ENABLE the input relay or actual input signal of which status is read by the method of (1).

As soon as you disable an input relay or coil, the input relay becomes independent of the actual signal and the coil of the ladder circuit, holding the ON or OFF status it has kept so far.

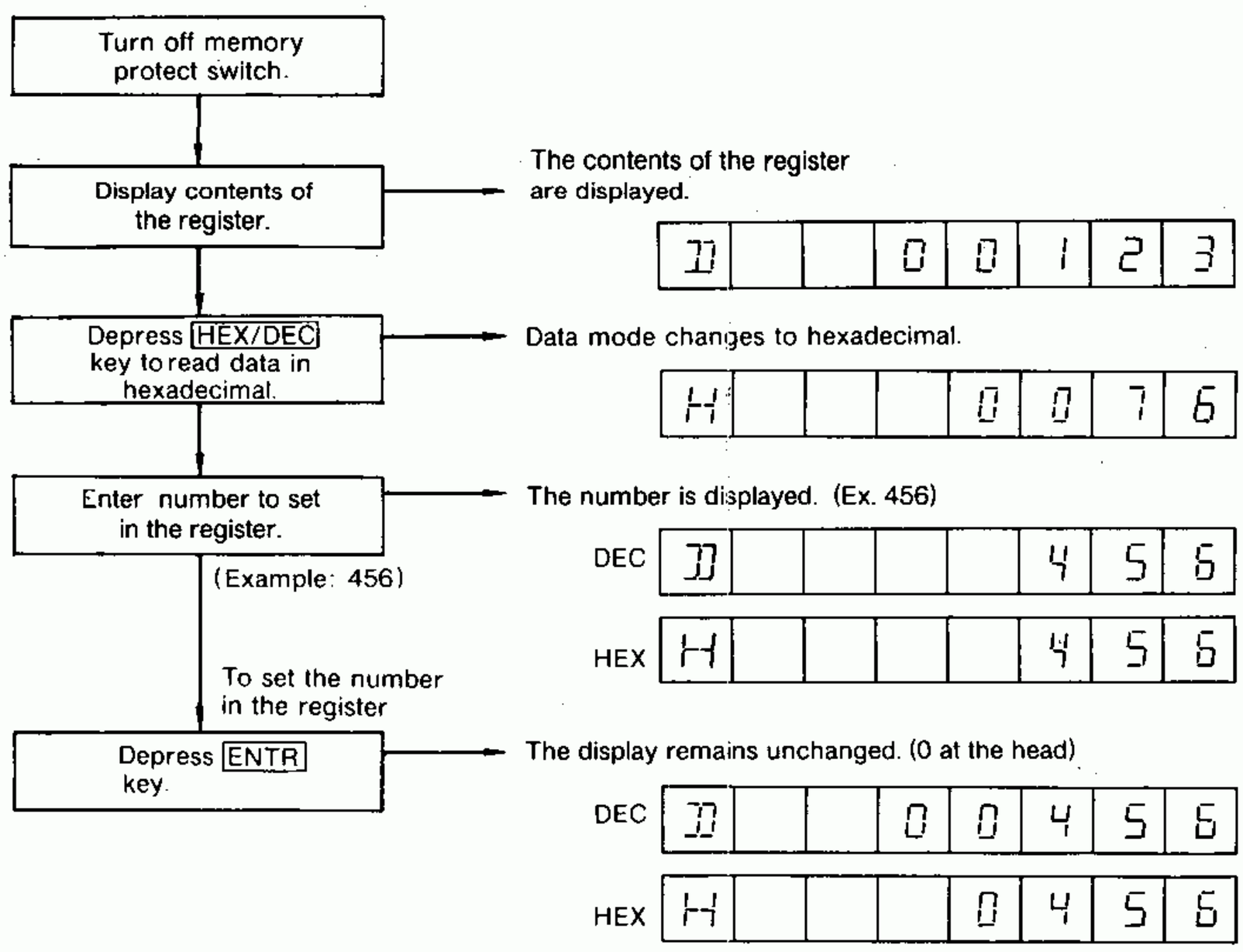
The disabled input relay and coil can be turned on and off unconditionally. When enabled, the input relay is activated according to the actual signal and the coil to the logic of the ladder circuit.



- DSBL／ENBL and ON／OFF , change mode alternately when depressed.
- The reference number is increased (decreased) by one every time NEXT (PREV) is depressed.

(4) Setting or Altering of Constant Register, Holding Register

It is possible to set a value in the constant register , holding register of which contents are read by the method of (2).



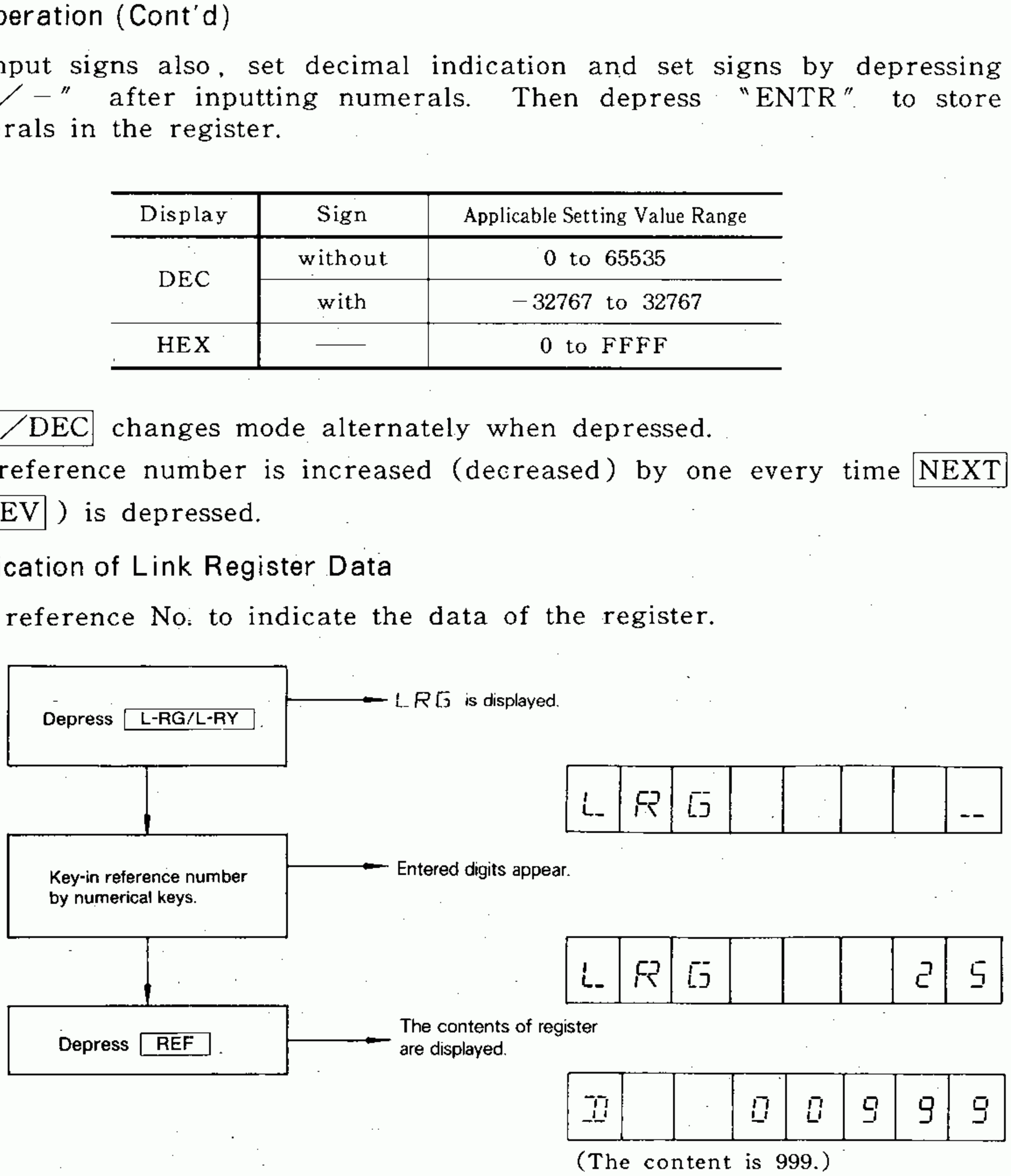
- To input signs also, set decimal indication and set signs by depressing "+/−" after inputting numerals. Then depress "ENTR" to store numerals in the register.

Display	Sign	Applicable Setting Value Range
DEC	without	0 to 65535
	with	−32767 to 32767
HEX	——	0 to FFFF

- HEX/DEC changes mode alternately when depressed.
- The reference number is increased (decreased) by one every time NEXT (PREV) is depressed.

(5) Indication of Link Register Data

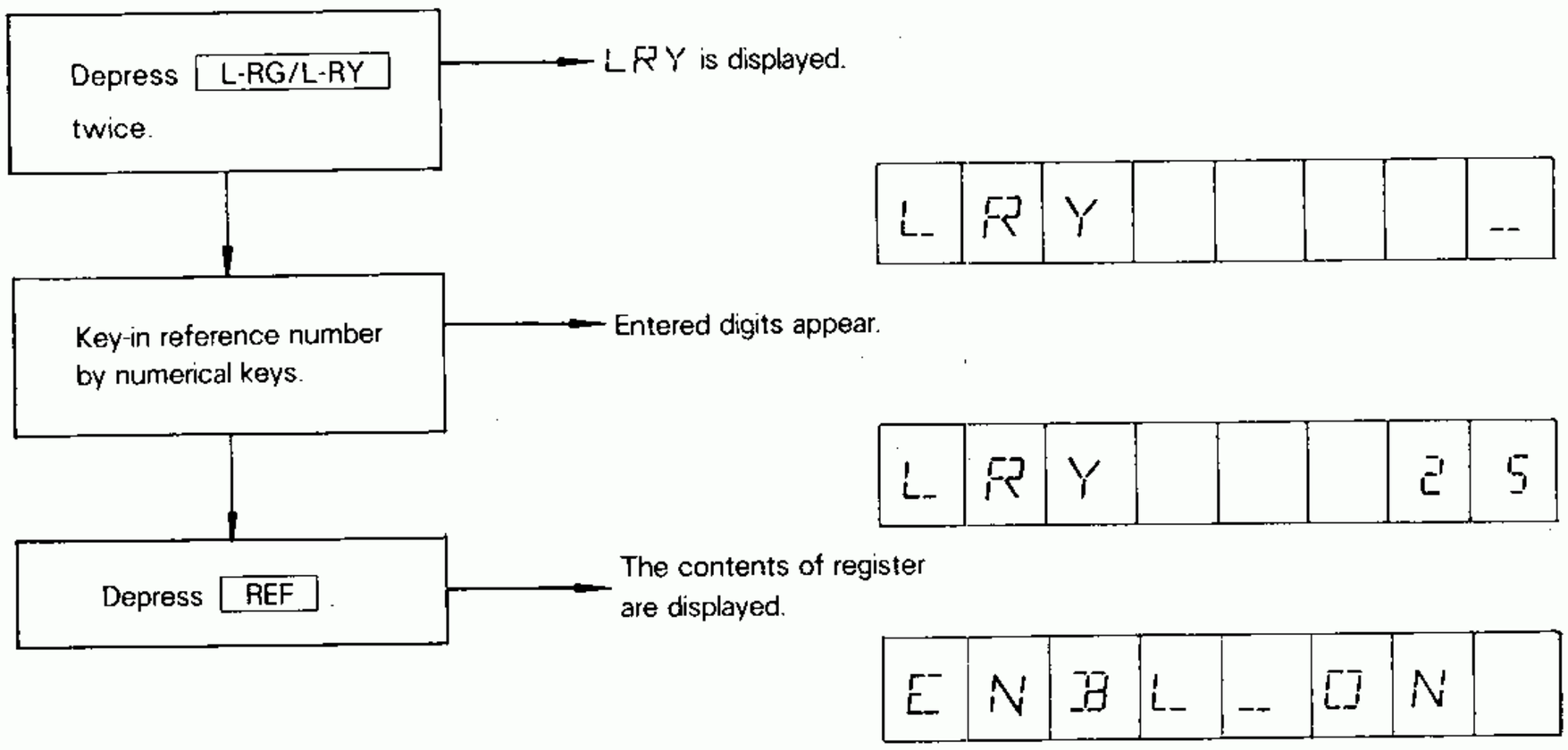
Set the reference No. to indicate the data of the register.



- Data is updated each scan.
- Reference No. is increased (decreased) by 1 each time NEXT (PREV) key is depressed.
- HEX/DEC changes mode alternately when depressed.
- The contents of link register cannot be set or changed.

(6) Status Indication of Link Coil (Relay)

Set the reference No. to indicate the link coil (relay).

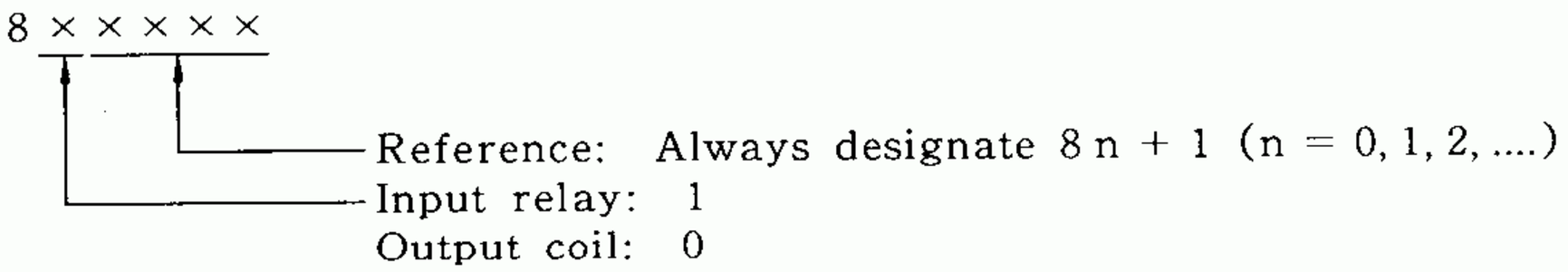


- The data will be refreshed in every scanning cycle.
- The reference number is increased (decreased) by one every time NEXT (PREV) is depressed.
- The contents of link coil (relay) cannot be set or changed.

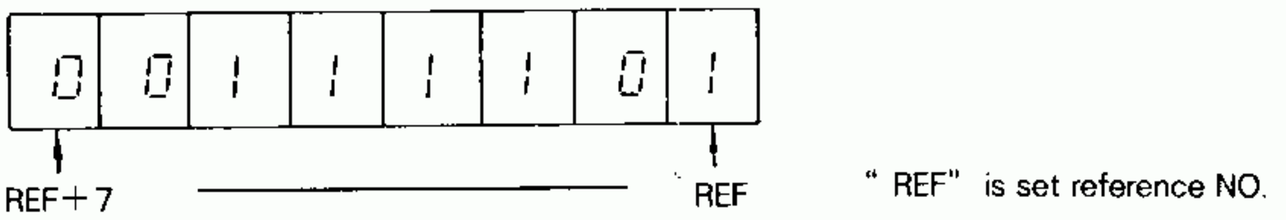
(7) 8-point Indication of Input Relay and Output Coil Status

Data of eight references in succession can be indicated only with input relays and output coils by inputting specified 6-digit No.

- Operate as in (1). The specified No. will be as follows. "8" is fixed.



- ON and OFF state is indicated as follows:



- Reference No. is increased (decreased) by 8 each time "NEXT" ("PREV") is depressed.

(8) Displaying Port Parameters

It is possible to display a communication parameter between the communication module and a device (the programming panel, etc.) connected to a port by entering a special number which is described below. 6 and 000 are fixed.

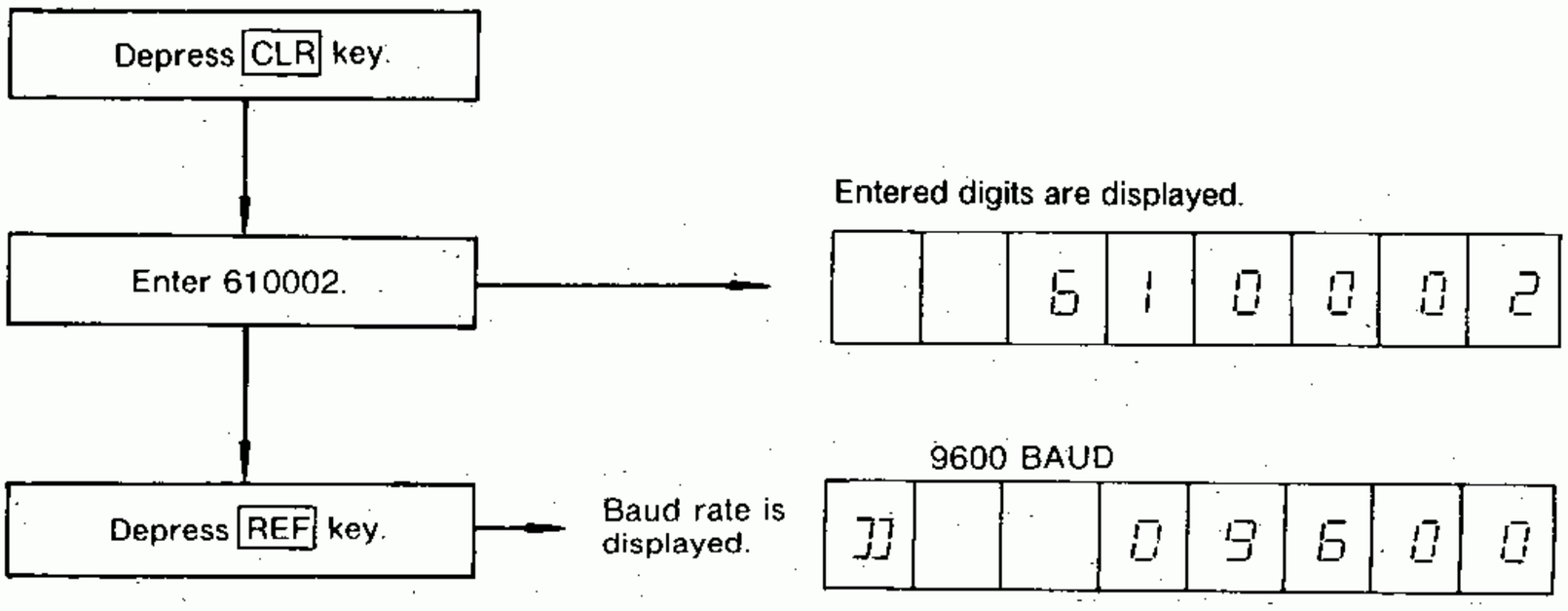
6X000Y

Parameter

Port Number
X = 1 Port 1
X = 3 Port 3
X = 4 Port 4

Y = 1 Device address
2 Baud rate
3 Use of parity
4 Parity when used
5 Number of stop bits
6 Communication mode
7 Preset value of port delay timer

Operate as follows to look at the baud rate of the device connected to port 1, for example.

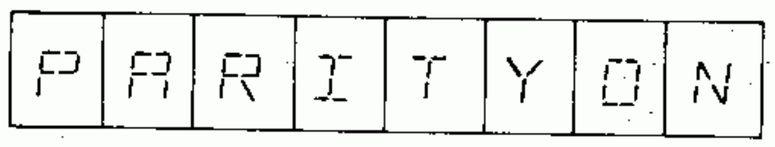


Communication parameters associated with Y are as follows.

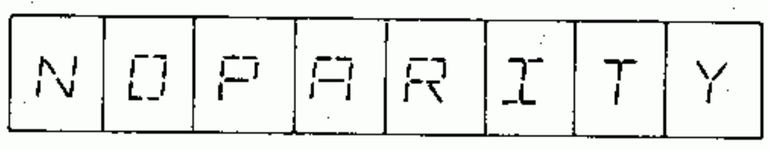
Y = 1 Device address: 1 to 247

Y = 2 Baud rate: 150, 300, 600, 1200, 2400, 4800, 9600, or 19200

Y = 3

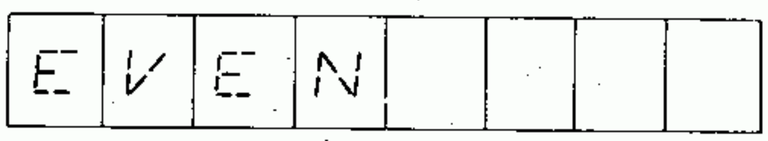


• • Parity check performed.

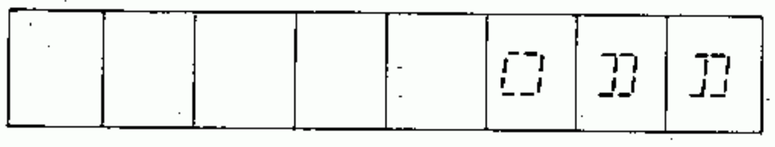


• • Parity check not performed.

Y = 4



• • Even parity.

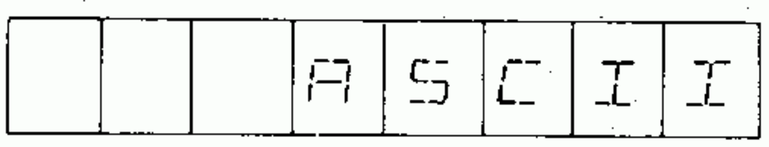


• • Odd parity.

Y = 5 Number of stop bits: 1 or 2.

Y = 6 Communication mode

• • RTU



• • ASCII

Y = 7 Preset value of port delay timer (GL40S delays response by this value): 0 to 255 (correspond to 0 to 2550 msec)

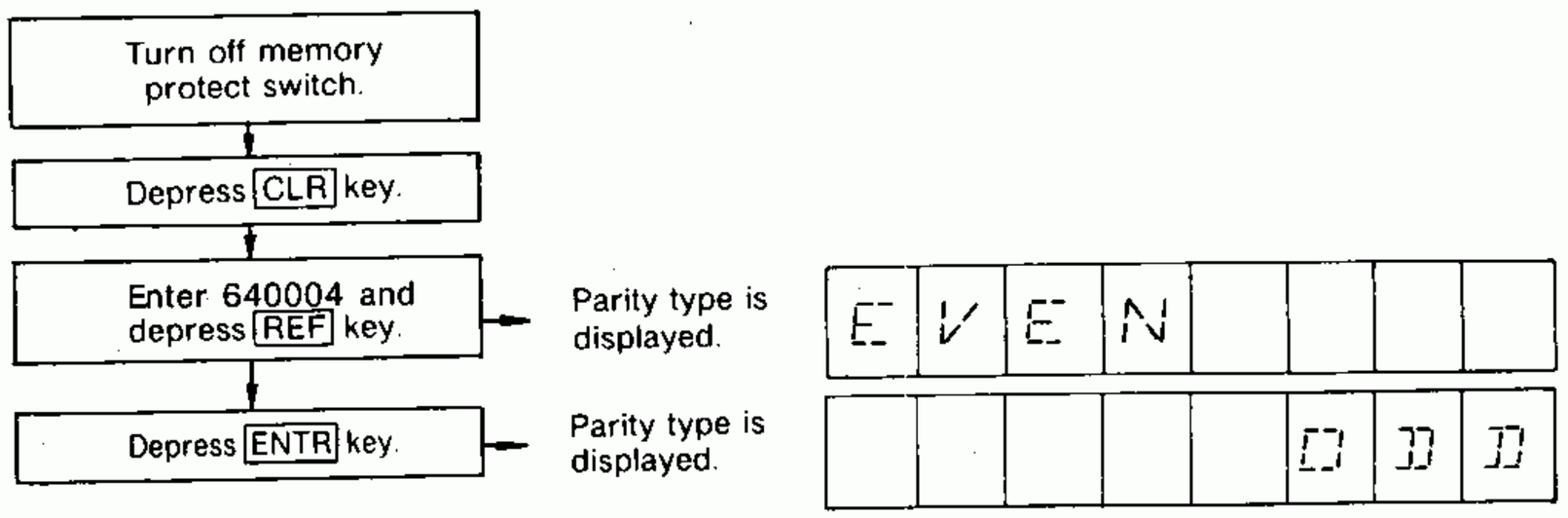
- The value of Y is increased (decreased) by one every time NEXT (PREV) is depressed.
- Device address, baud rate and delay timer are displayed only with unsigned decimal.

(9) Setting or Altering of Communication Parameters

To set a device address or baud rate, enter the number to be set and push [ENTR], then the number is set. The parameters of port 1 are fixed except device address. The device address of port 1 can be set from 1 to 99 by rotary switch.

In other cases, push [ENTR] only. Then one of two choices changes to the other.

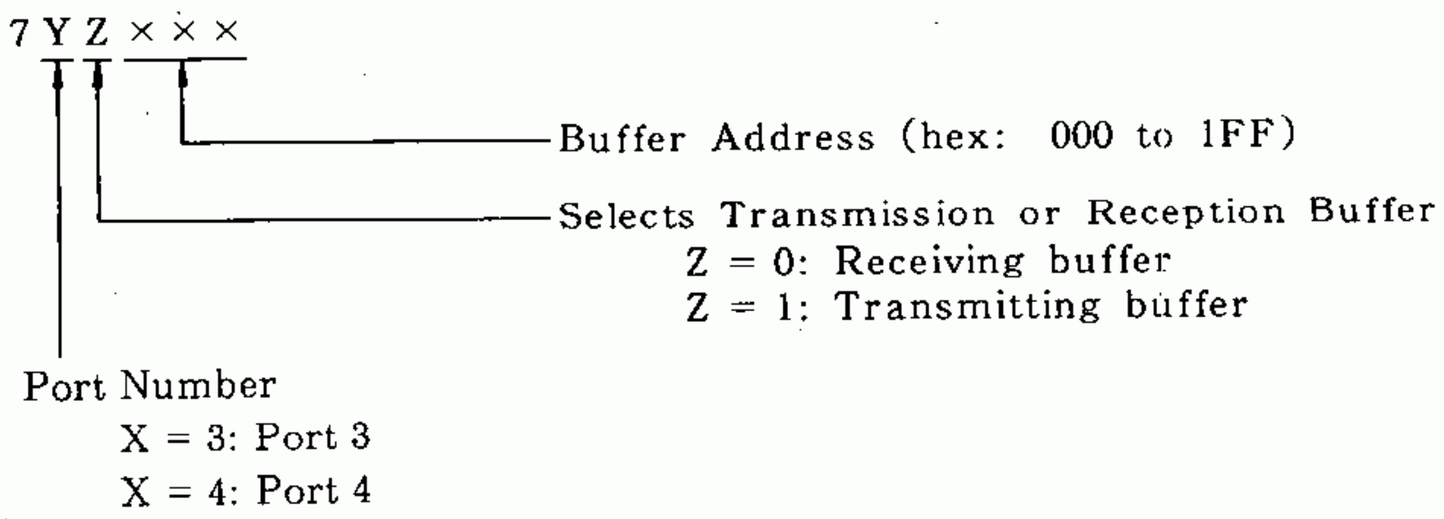
To change even parity of port 4 to odd, for example;



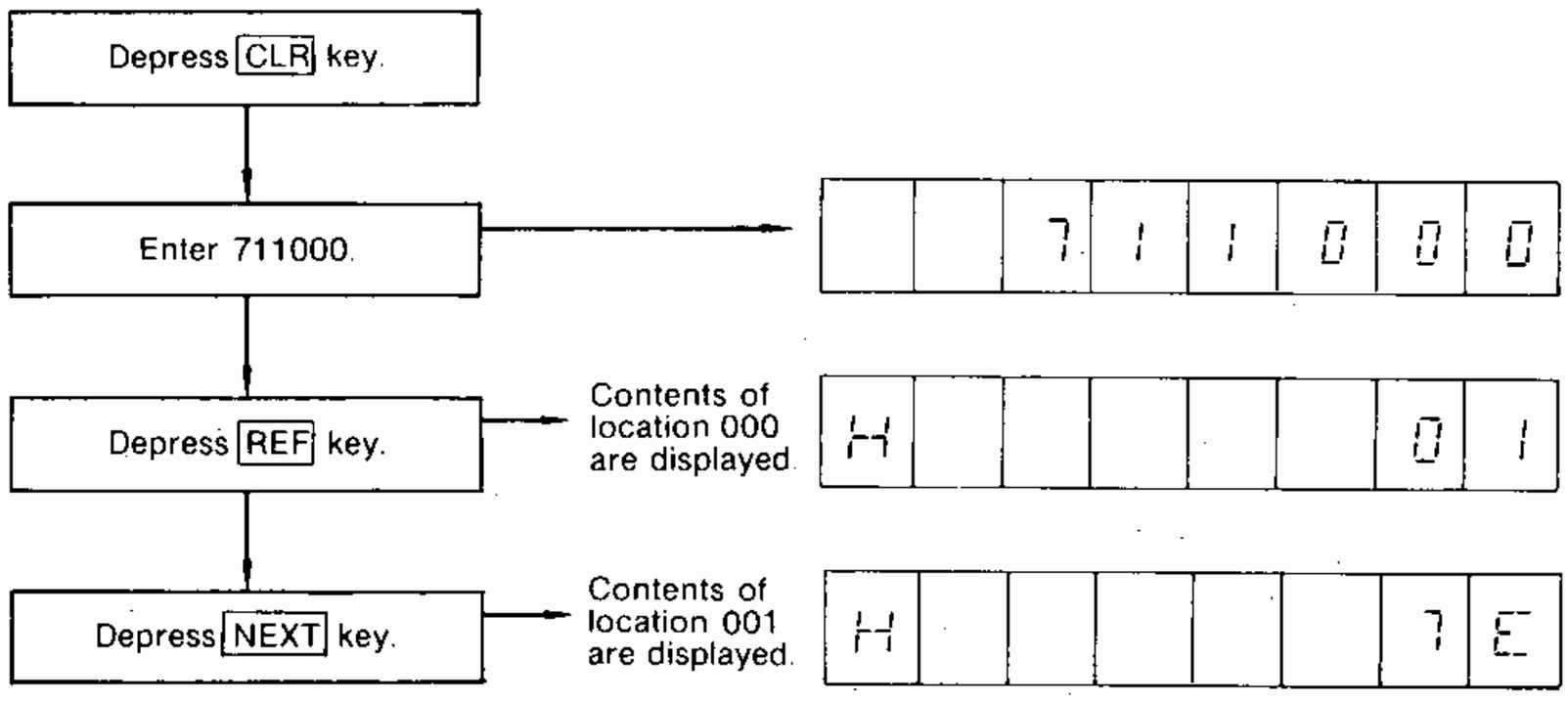
- The value of Y is increased (decreased) by one every time [NEXT] ([PREV]) is depressed.

(10) Displaying of Contents of Communication Buffer

As in communication parameters, it is possible to display the contents of the communication buffer of a MEMOBUS port by entering a special number which is described below. 7 is fixed.



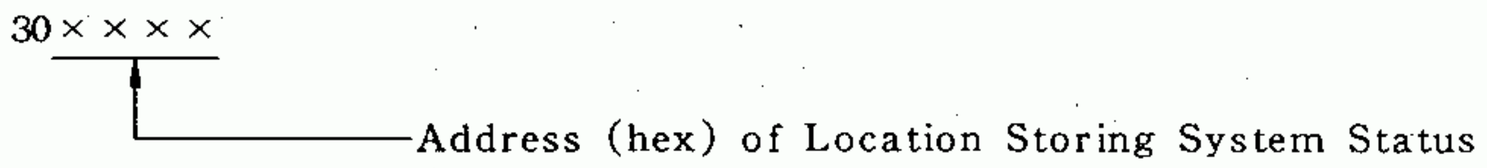
Operate as follows to look at the contents of transmitting buffer of port 1, for example.



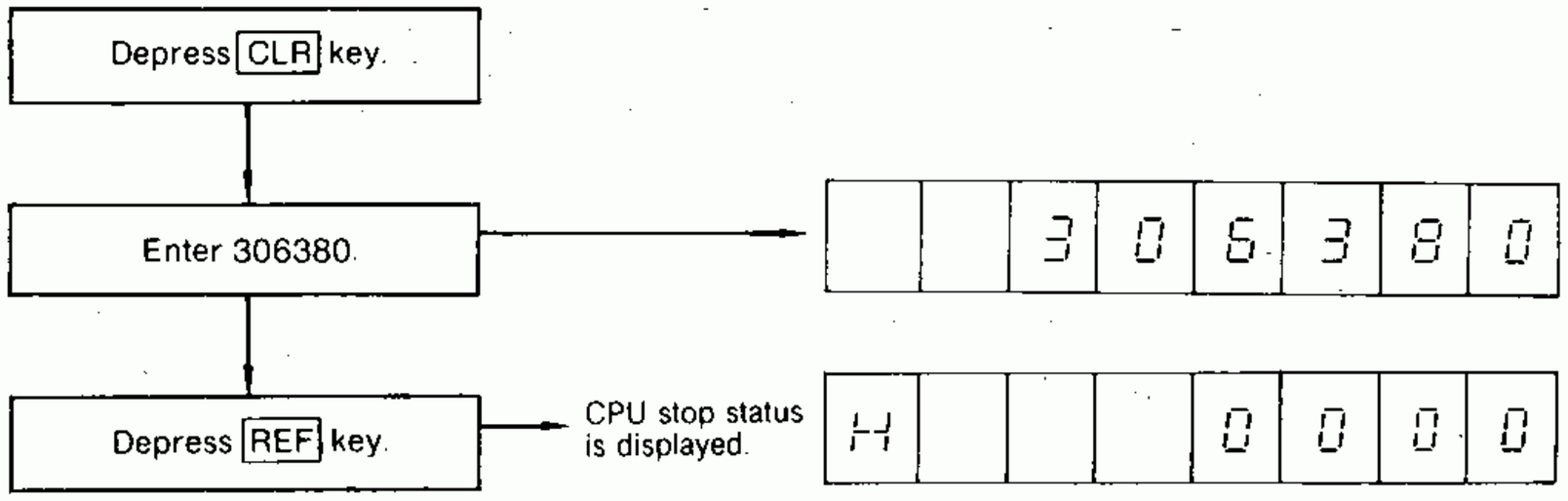
- The data is displayed in hex form at first.
- The buffer address × × × × is increased (decreased) by one every time NEXT (PREV) is depressed.
- **Transmission/reception buffer contents of port 1 cannot be displayed from RAP.**

(11) Displaying of GL40S System Status

It is possible to display the GL40S system status by entering a special number which is described below. 3 and 0 are fixed.



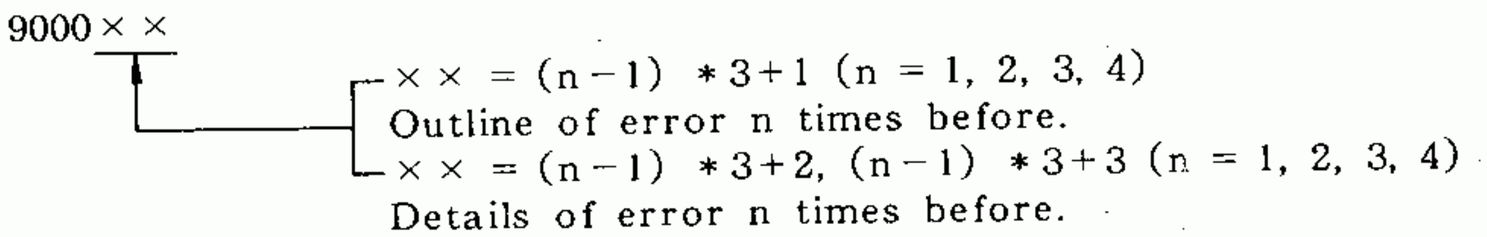
GL40S status storing address numbers are described in Par. 9.6.
Operate as follows to read the CPU stop status, for example.



- **The status is displayed in hex form at first.**
- The address × × × × is increased (decreased) by one every time NEXT (PREV) is depressed.

(12) Failure History State Display

Input the following specified No. to indicate the states of past four CPU stoppages.



9.5.3 Error Codes

(1) Wrong Operation

Table 9.8 summarizes the codes indicating wrong operations. Wrong data would not be stored in memory, but the normal state can be restored by recovery operation. Note that no error code will be displayed on RAP if a similar mistake is made on the programming panel: it appears only when RAP is operated.

Table 9.8 Error Codes of Wrong Operations

Error Code	Error Description
ERROR_01	Reference is not correct.
ERROR_02	Expansion memory (for future) is not provided.
ERROR_03	For future expansion.
ERROR_04	Setting and changing data are out of range.
ERROR_05	Input relay, coil are not DISABLE.
ERROR_06	Setting and changing are not permitted.
ERROR_07	Memory protect is switched on.
ERROR_08	Other wrong operations.

- Return-home method

Depress CLR to indicate data or to return to indicate data if error occurred during data setting. In other cases, indication is cleared.

Note 'CLR not only clears the digital and character displays to 0 but also cancels the function specified. Before making a new entry, it is recommended to depress CLR twice in order to cancel the data and function entered.

(2) System Errors

Table 9.9 System Errors

Error Code	Content
SYSERR 10	Transmission time-out error with CPU module
SYSERR 20	Transmission time-out error with IOP module
SYSERR 99	ROM total sum error of RAP

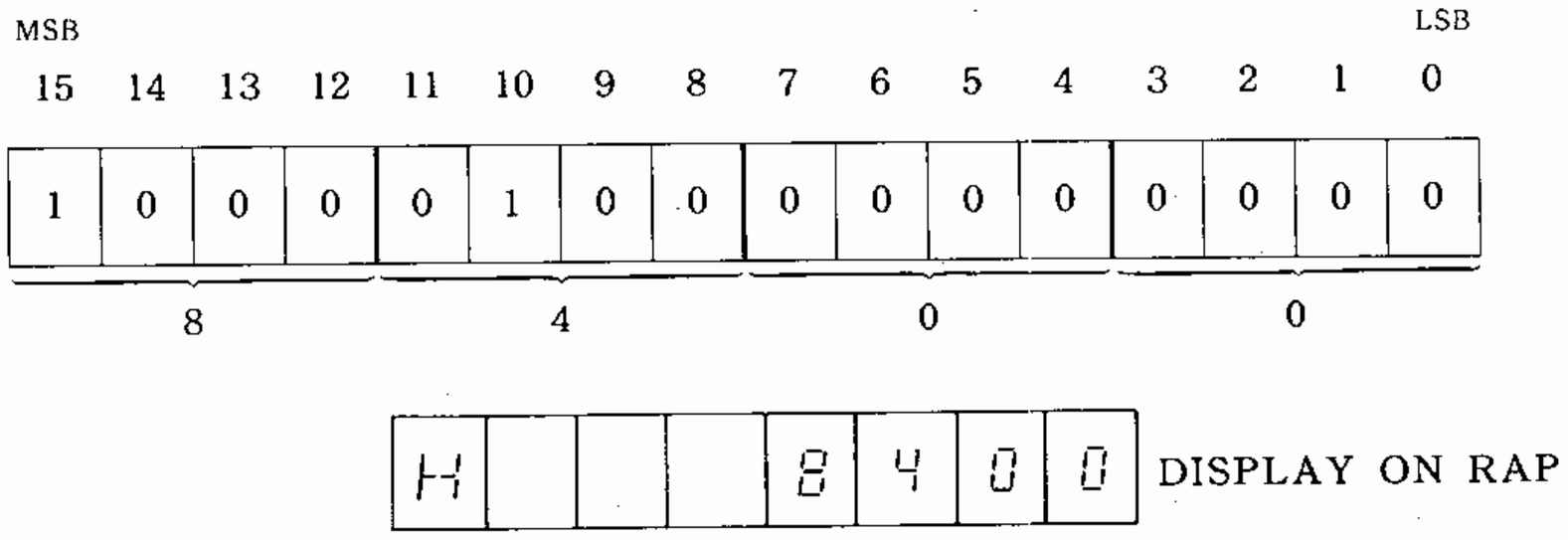
9.5.4 Status LED Lamp

The LED lamp located at lower right of the display panel is lit if the following occurs:

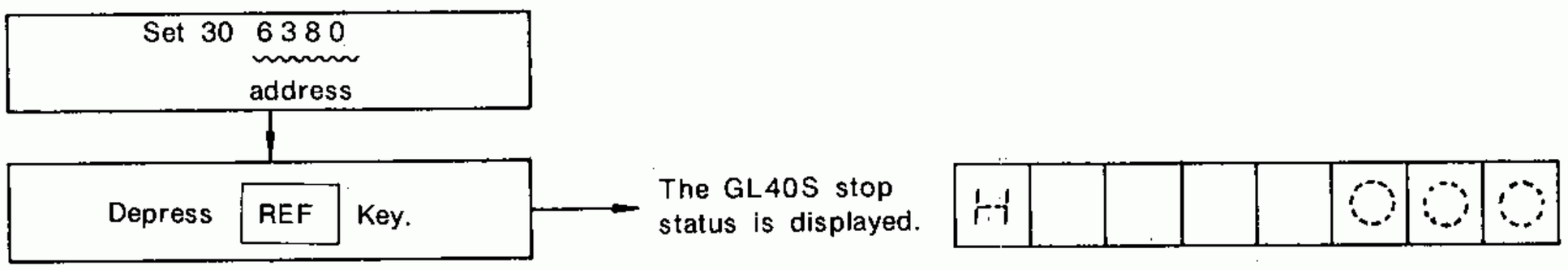
- The relay switches on during indication of the input relay data. (Except for 8-point simultaneous indication)
- The coil switches on during indication of the output coil data. (Except for 8-point simultaneous indication)
- The link relay switches on during display of the link relay data.

9.6 GL40S SYSTEM STATUS

The GL40S system status is stored in fixed address of the CPU memory and can be indicated on RAP. The system status is indicated by bit patterns or codes. A 16-bit binary data can be displayed in hex form as follows.



Example 1: To display the GL40S stop status (address 6380) on RAP



Example 2: To display the system status (It is displayed if an error occurs.)

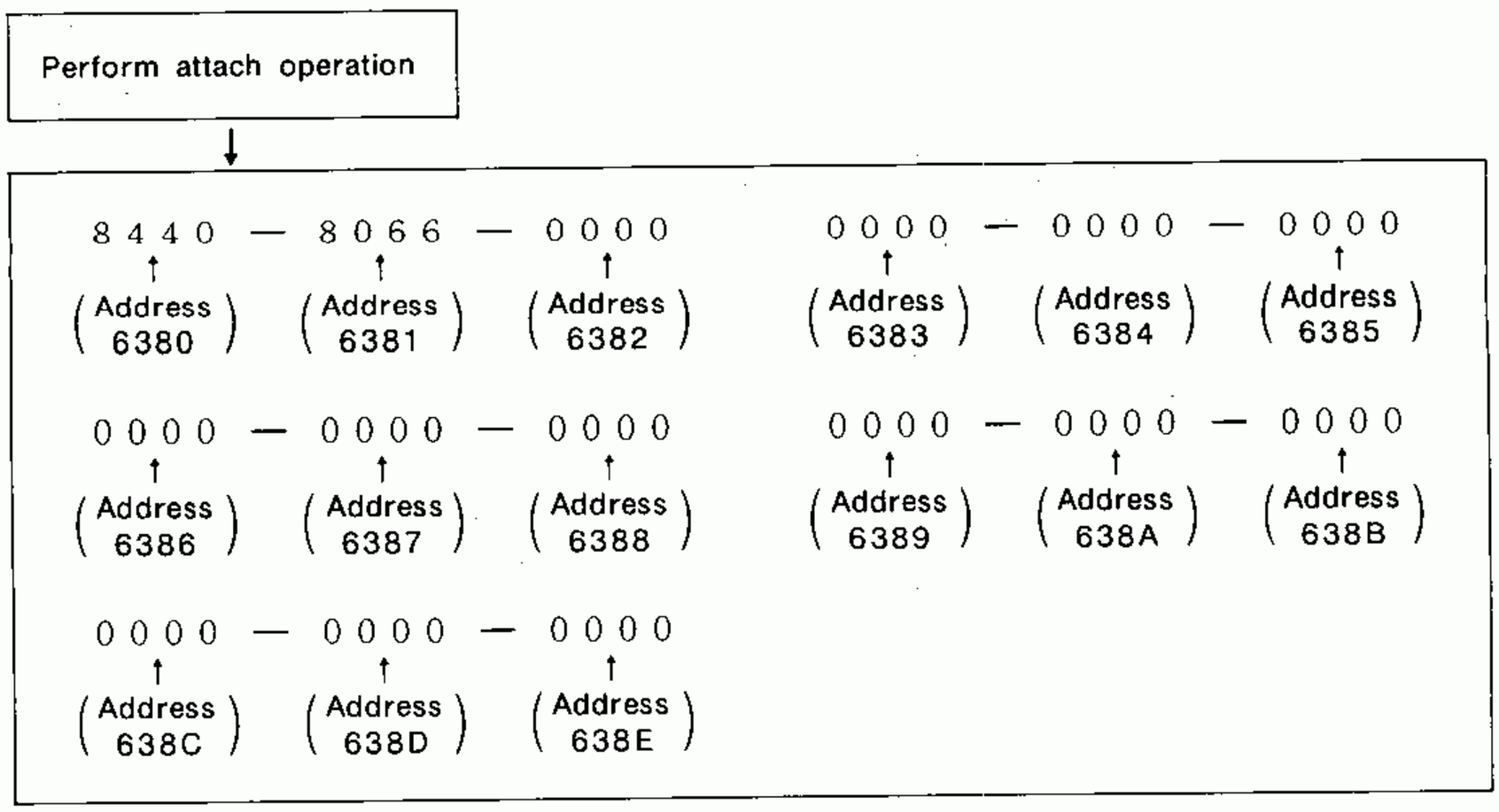


Table 9.10 shows the GL40S system status. The GL40S system status is stored in the table at a length of 90 from address 6380 to address 640F. Addresses 63B2 to 640F are for future expansions and do not store any information at present.

The system status can be checked anytime using the arithmetic function STAT mentioned in Par. 5.9.11 in addition to being displayed on the RAP.

Table 9.10 GL40S System Status

ADDRESS	
6380	GL40S STOP STATUS
6381	TOTAL SUM ERROR
6382	DATA ERROR
6383	GL40S STOP STATUS AT LAST POWER CUT-OFF
6384	TOTAL SUM ERROR AT LAST POWER CUT-OFF
6385	DATA ERROR AT LAST POWER CUT-OFF
6386	GL40S STOP STATUS OF ONE POWER CUT-OFF BEFORE THE LAST
6387	TOTAL SUM ERROR OF ONE POWER CUT-OFF BEFORE THE LAST
6388	DATA ERROR OF ONE POWER CUT-OFF BEFORE THE LAST
6389	GL40S STOP STATUS OF TWO POWER CUT-OFF BEFORE THE LAST
638A	TOTAL SUM ERROR OF TWO POWER CUT-OFF BEFORE THE LAST
638B	DATA ERROR OF TWO POWER CUT-OFF BEFORE THE LAST
638C	GL40S STOP STATUS OF THREE POWER CUT-OFF BEFORE THE LAST
638D	TOTAL SUM ERROR OF THREE POWER CUT-OFF BEFORE THE LAST
638E	DATA ERROR OF THREE POWER CUT-OFF BEFORE THE LAST
638F	FOR FUTURE EXPANSION
6390	GL40S MACHINE STATUS
6391	FOR FUTURE EXPANSION
6392	COMM STATUS
6393	PC-LINK STATUS
6394	FOR FUTURE EXPANSION
6395	SERVO I/F 1 STATUS
6396	SERVO I/F 2 STATUS
6397	SERVO I/F 3 STATUS
6398	SERVO I/F 4 STATUS
6399	SCAN TIME (unit : 10ms)
639A	FOR FUTURE EXPANSION
639B	FOR FUTURE EXPANSION
639C	FOR FUTURE EXPANSION
639D	FOR FUTURE EXPANSION
639E	FOR FUTURE EXPANSION
639F	FOR FUTURE EXPANSION
63A0	LOCAL I/O ERROR (RACK 1)
63A1	LOCAL I/O ERROR (RACK 2)
63A2	LOCAL I/O ERROR (RACK 3)
63A3	LOCAL I/O ERROR (RACK 4)
63A4	FOR FUTURE EXPANSION
63A5	LOCAL RACK ERROR STATUS
63A6	LOCAL RACK ACK ERROR STATUS
63A7	LOCAL RACK ACK ERROR HISTORY
63A8	FOR FUTURE EXPANSION
63A9	FOR FUTURE EXPANSION
63AA	FOR FUTURE EXPANSION
63AB	FOR FUTURE EXPANSION
63AC	FOR FUTURE EXPANSION
63AD	FOR FUTURE EXPANSION
63AE	FOR FUTURE EXPANSION
63AF	FOR FUTURE EXPANSION
63B0	PC-LINK STATION STATUS
63B1	PC-LINK STATION STATUS
⋮	FOR FUTURE EXPANSION
640F	FOR FUTURE EXPANSION

To see its content, set 30XXXX on RAP and depress [REF] key.
address

(1) The following names for individual bits are set in explaining the semantics of the status.

· CPU stop status

	MSB															LSB
6380	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

· Table total sum error

	MSB															LSB
6381	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

· Table data error

	MSB															LSB
6382	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

· GL40S machine status

	MSB															LSB
6390	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

· Local rack status

	MSB															LSB
63A5 to 63A7	G15	G14	G13	G12	G11	G10	G9	G8	G7	G6	G5	G4	G3	G2	G1	G0

· Link station status

	MSB															LSB
63B0	H31	H30	H29	H28	H27	H26	H25	H24	H23	H22	H21	H20	H19	H18	H17	H16

	MSB															LSB
63B1	H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1	H0

(2) GL40S Stop Status (Address 6380)

Table 9.11 GL40S Stop Status (Address 6380)

Bit	Semantics
B15	Set if either 1 or B14 to B0 is 1 in case CPU is stopping.
B14	1 if watchdog timer error.
B13	1 if real time clock error.
B12	1 if bit ALU error.
B11	1 if invalid memory allocation.
B10	1 if ladder circuit error (Set if C15, D15, D11, D10 or D6 is 1.)
B9	Not used. Size mismatched (ROM operation)
B8	Not used. No ROM (ROM operation)
B7	1 if subroutine circuit error (Set if C12, C4, D12, D7 or D3 is 1.)
B6	Not used.
B5	1 if user status table total sum error.
B4	1 if T-COP table error (Set if C11, C3 is 1.)
B3	Other allocated table error (Set if C10, C9, C8 or C7 is 1.)
B2	Not used.
B1	PC link allocated table total sum error.
B0	Subsystem trouble

(3) Total Sum Errors (Address 6381)

Table 9.12 Total Sum Errors

Bit	Semantics
C15	Ladder circuit total sum error
C14	Not used.
C13	Not used.
C12	Subroutine circuit total sum errors
C11	T-COP table total sum error
C10	Not used.
C9	Not used.
C8	User memory allocated table total sum errors
C7	Transmission parameter table total sum errors
C6	Not used.
C5	Not used.
C4	Subroutine entry table total sum errors
C3	T-COP entry table total sum errors
C2	Not used.
C1	Not used.
C0	Not used.

(4) Data Errors (Address 6382)

Table 9.13 Data Error

Bit	Semantics
D15	Ladder circuit EOL error
D14	Not used.
D13	Not used.
D12	Subroutine circuit EOL error
D11	Ladder circuit EOS error
D10	Ladder circuit SON error
D9	Not used.
D8	Not used.
D7	Subroutine circuit SOS error
D6	Not used.
D5	Not used.
D4	Not used.
D3	Not used.
D2	Not used.
D1	Not used.
D0	Not used.

(5) GL40S Machine Status (Address 6390)

Table 9.14 GL40S Machine Status

Bit	Semantics	Bit	Semantics
E15	For future expansion	E7	Not used.
E14	Not used.	E6	Not used.
E13	Not used.	E5	Not used.
E12	Not used.	E4	Not used.
E11	Not used.	E3	Not used.
E10	COMM error	E2	Not used.
E9	Link error	E1	Not used.
E8	Not used.	E0	Not used.

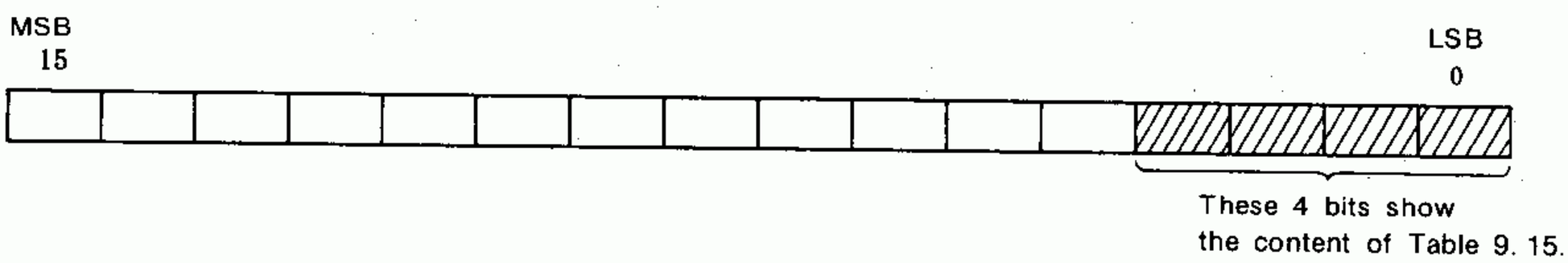
(6) COMM, PC-LINK, SERVO I/F Status

These status data are read out from I/F RAM of each module. If status data are error codes, corresponding bit in machine status is set at "1".

Table 9.15 COMM, PC-LINK, SERVO I/F Status

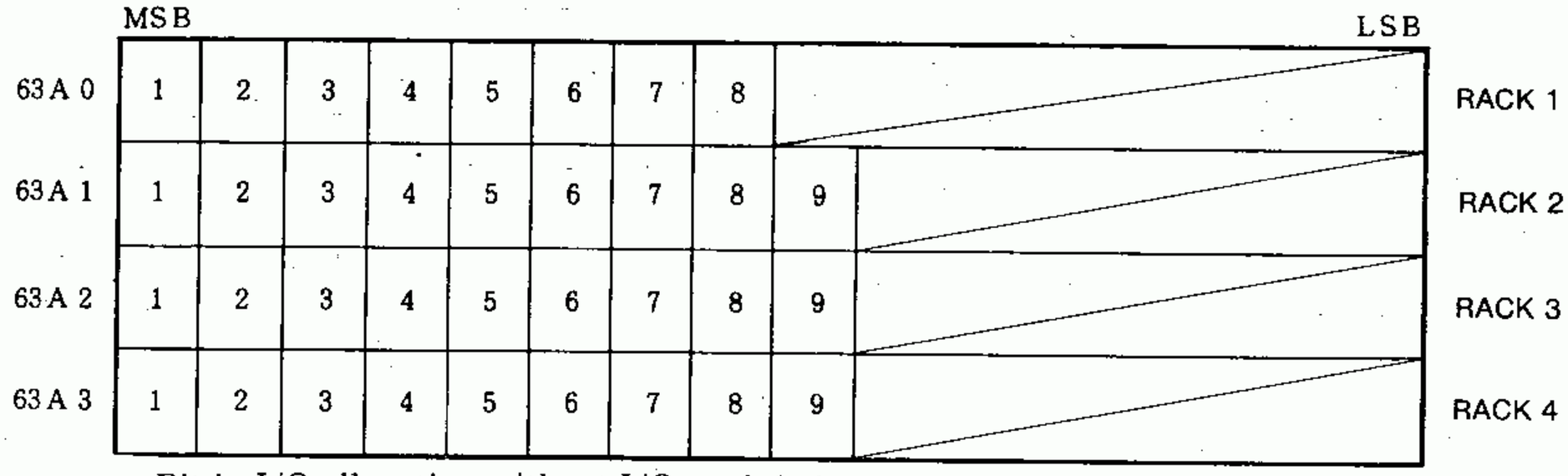
CODE (H)	Meaning
0	Normal
1	ROM error
2	RAM error
3	Common memory error
8	PC-LINK stop (For only PC-LINK status)
F	Watchdog error

COMM : address 6392
PC-LINK : address 6393
Servo I/F 1 : address 6395
Servo I/F 2 : address 6396
Servo I/F 3 : address 6397
Servo I/F 4 : address 6398



(7) Local I/O Error (Address 63A0 to 63A3)

Data to show errors of modules mounted in the local I/O rack are stored. Fig. 9.12 shows relationships between bits and slots. Numerals indicate slot Nos.



Bit in I/O allocation without I/O module is set to "1".

Fig. 9.12 Local I/O Error Table

(8) Local RACK status

Data to show error of each local I/O rack are stored.

Table 9.16 Local Rack Status

Bits	Semantics		
	Local Rack Error Status Address 63A5	Local Rack ACK Error Status Address 63A6	Local Rack History ACK Error Status Address 63A7
G15 to G4	Not used.	Not used.	Not used.
G3	"1" if rack 4 Bus check error.	"1" if no response of rack 4	"1" if no response of rack 4 at last scan
G2	"1" if rack 3 Bus check error.	"1" if no response of rack 3	"1" if no response of rack 3 at last scan
G1	"1" if rack 2 Bus check error.	"1" if no response of rack 2	"1" if no response of rack 2 at last scan
G0	"1" if rack 1 Bus check error.	"1" if no response of rack 1	"1" if no response of rack 1 at last scan

(9) PC-LINK Station Status (Address 63B0, 63B1)

For each LINK station (32 stations max), CPU module stores data to show module-mounted condition and errors. Bit in normal station is set to "1." Bit in troubled station (including station with no module) is set to "0."

Table 9.17 LINK Station Status

Bits	Stations	Bits	Stations
H31	Station 16	H15	Station 32
H30	Station 15	H14	Station 31
H29	Station 14	H13	Station 30
H28	Station 13	H12	Station 29
H27	Station 12	H11	Station 28
H26	Station 11	H10	Station 27
H25	Station 10	H9	Station 26
H24	Station 9	H8	Station 25
H23	Station 8	H7	Station 24
H22	Station 7	H6	Station 23
H21	Station 6	H5	Station 22
H20	Station 5	H4	Station 21
H19	Station 4	H3	Station 20
H18	Station 3	H2	Station 19
H17	Station 2	H1	Station 18
H16	Station 1	H0	Station 17
Address 63B0		Address 63B1	

9.7 TROUBLESHOOTING

In order that repairs can be made as easily and quickly as possible without a knowledge of details, the GL40S is maintained in basic units of modules, that is, by module replacement.

If a failure occurs, the first thing to do is to identify the problem accurately and follow the prescribed procedure for maintenance. Make use of the general GL40S troubleshooting flowcharts provided in this section.

Maintenance of the GL40S requires spare parts listed on Par. 8.4.

Note
1. The symbols used in the flow charts have the following meanings:
   ▭ : Action ⬭ : Teminal or comment ◇ : Decision ○: Connector
2. Power on/off operations are omitted from the flow charts. Normally, power will be turned off before a maintenance action, and turned on after the action has been completed.

(1) Power Supply Module Check

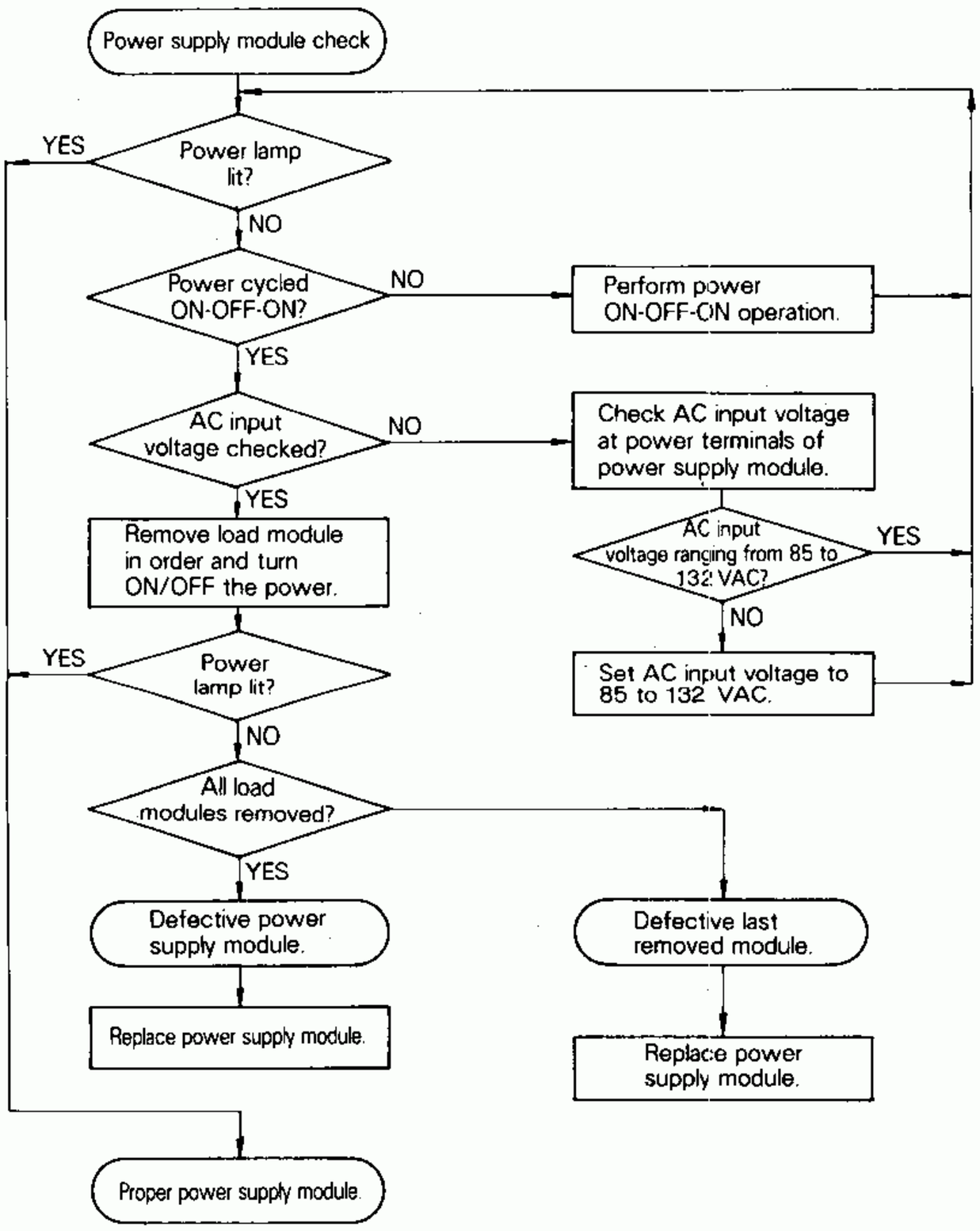


Fig. 9.13 Power Supply Module Check

Note
When circuit protector trips at checking PS21, reset the circuit protector for normal operation.

(2) CPU Module Check

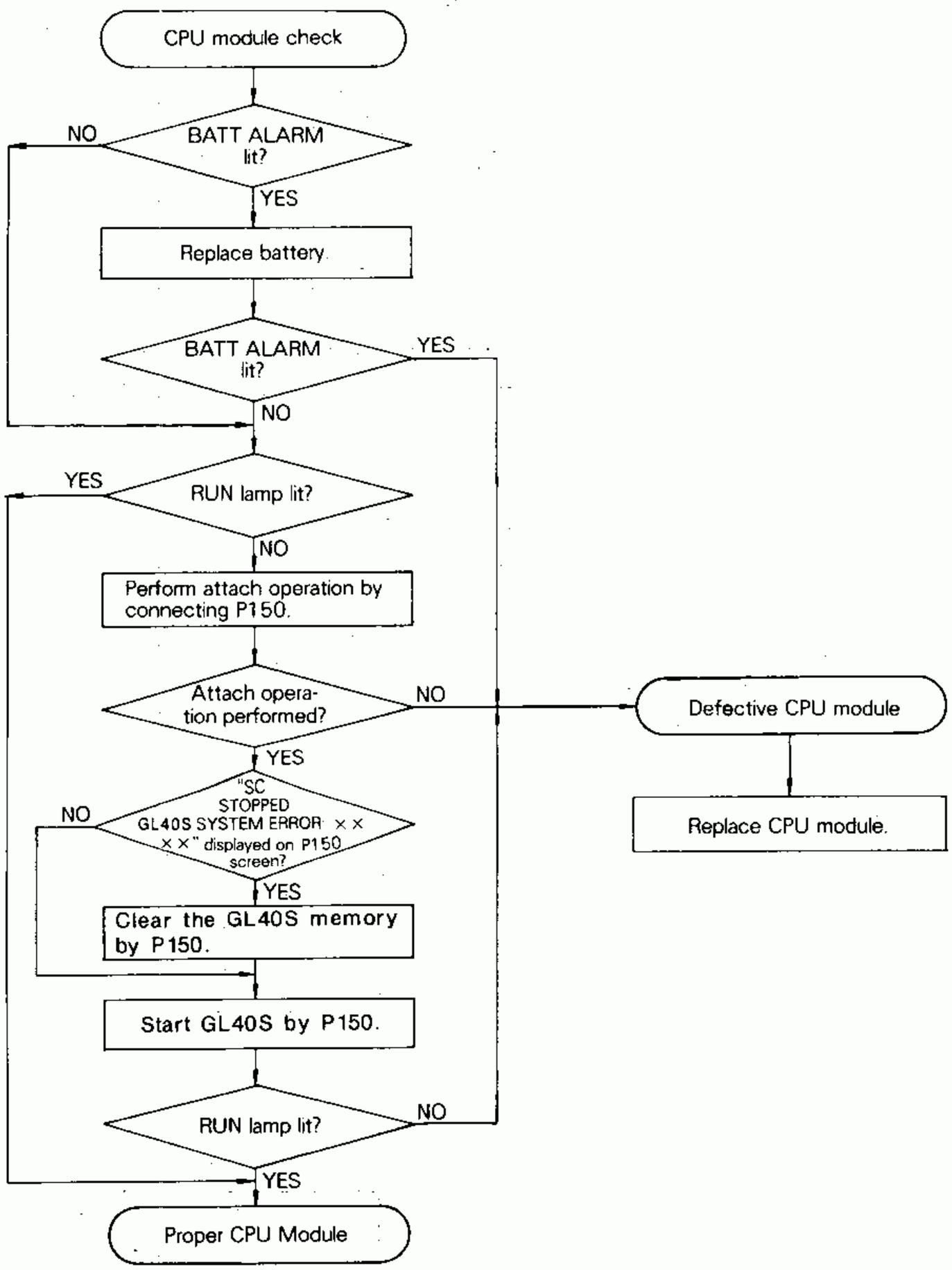


Fig. 9.14 CPU Module Check

(3) COMM Module Check

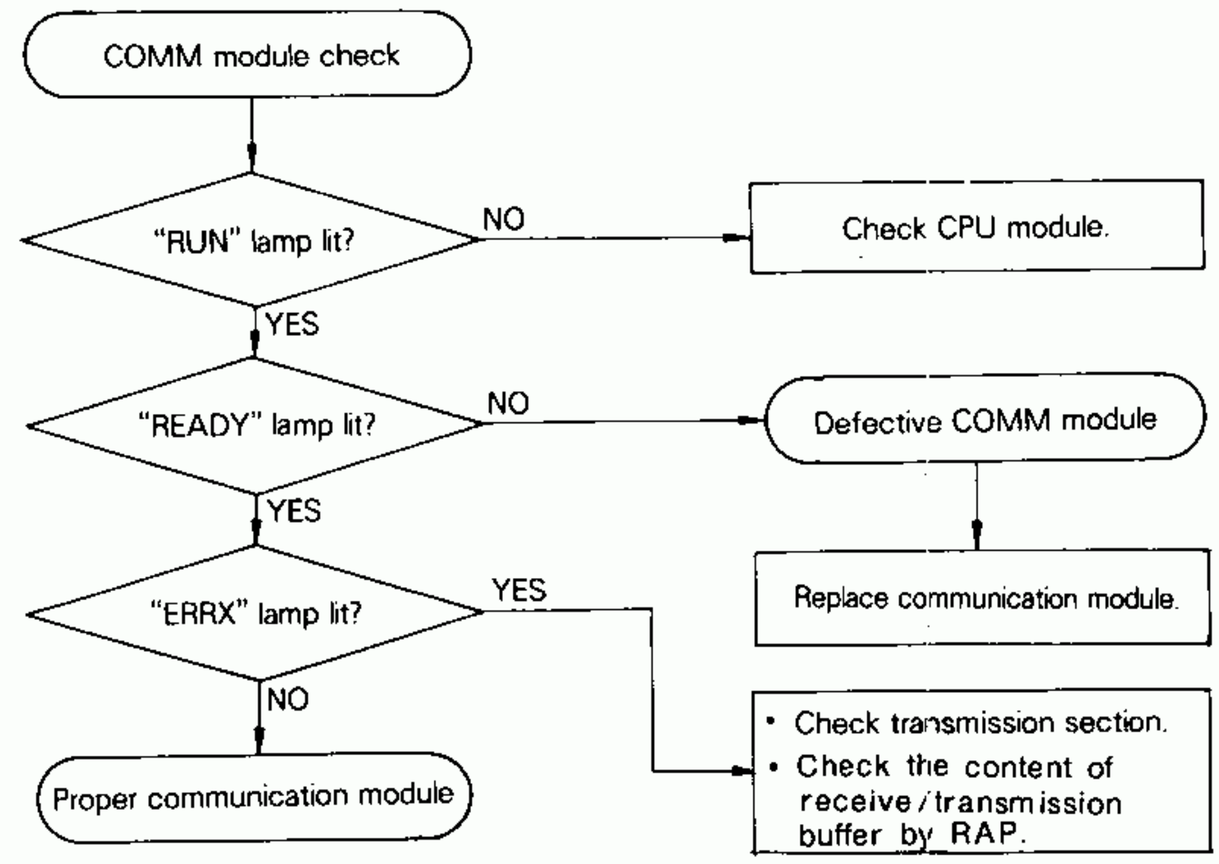


Fig. 9.15 COMM Module Check

(4) I/O Section Check

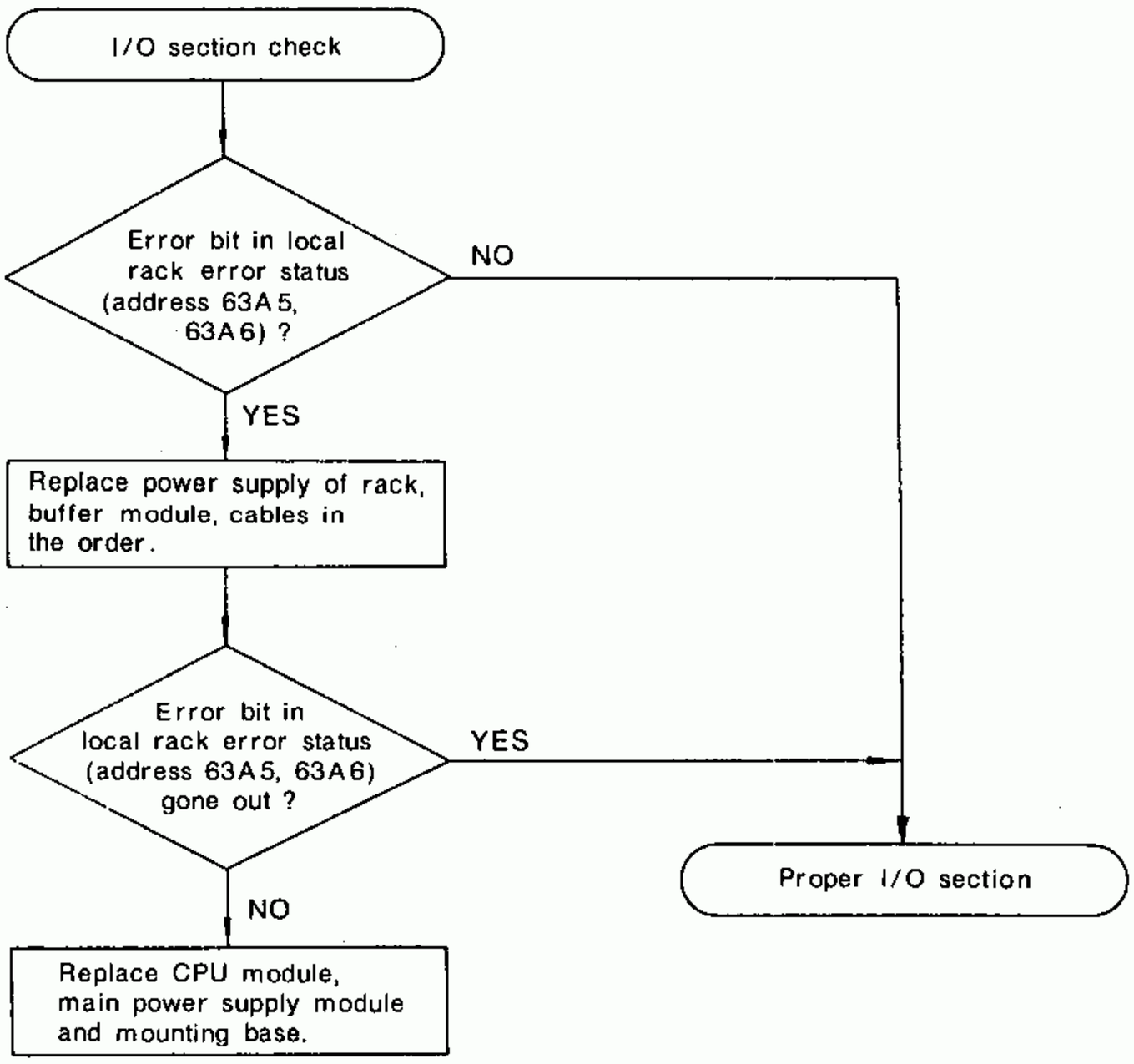


Fig. 9.16 I/O Section Check

(5) Input Module Check

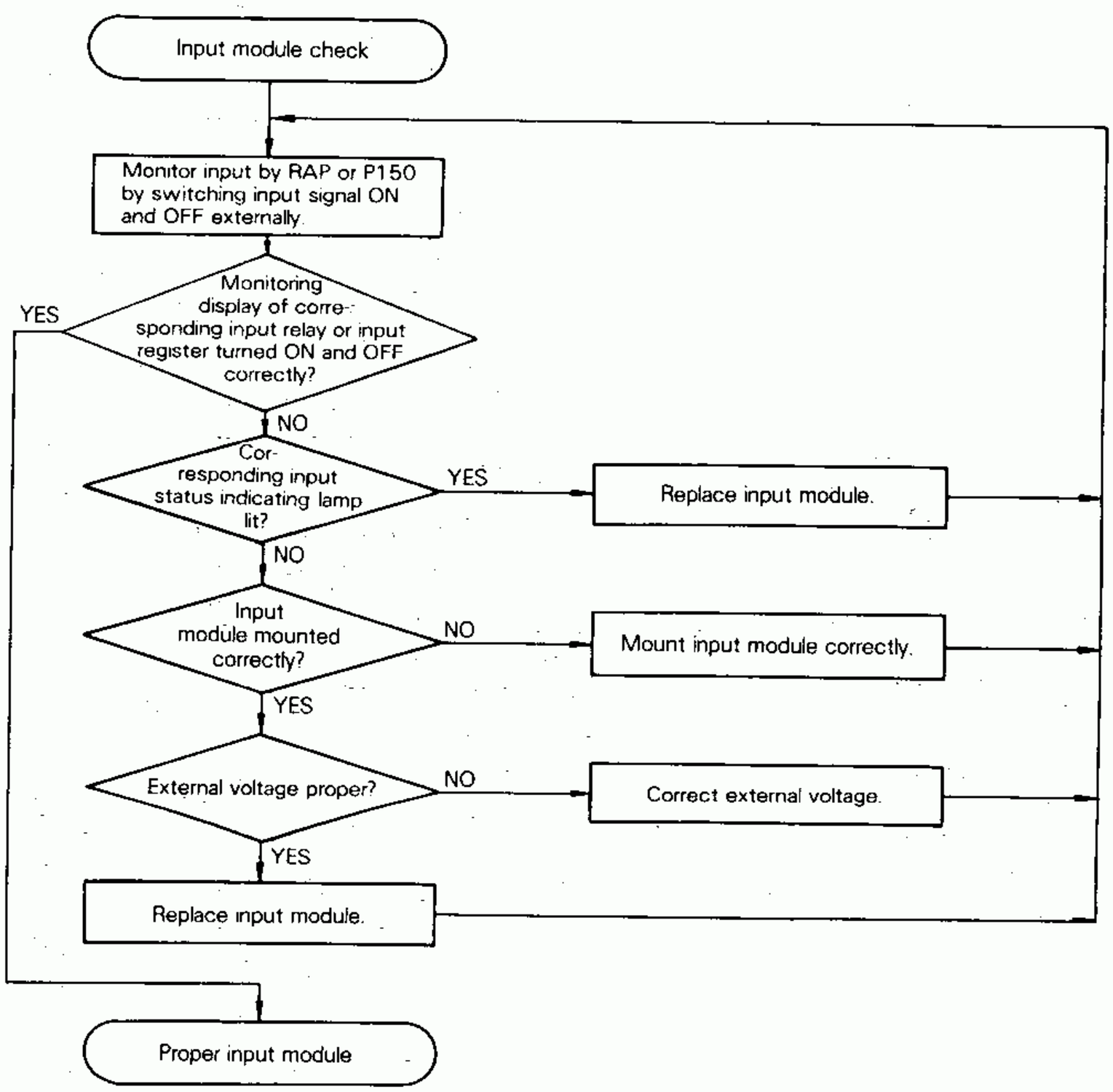


Fig. 9.17 Input Module Check

(6) Output Module Check

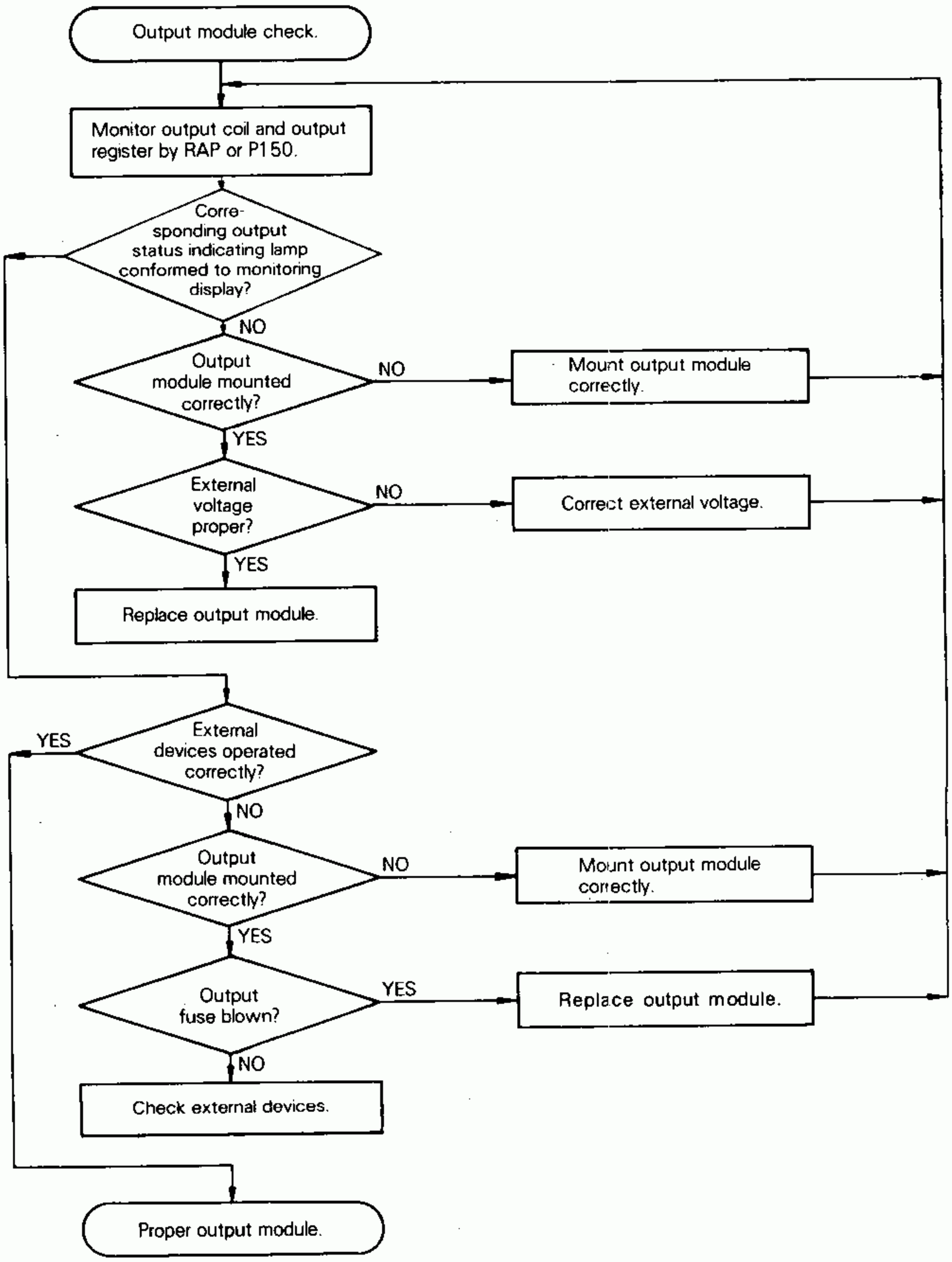


Fig. 9.18 Output Module Check

APPENDIX A

GL40S COMPONENTS LIST

Component	Type	Function or Application	Remarks
CPU Module	DDSCR-GL40S$_1$	Program memory 2k-word (24 bits/word)	
	DDSCR-GL40S$_2$	Program memory 4k-word (24 bits/word)	
	DDSCR-GL40S$_3$	Program memory 8k-word (24 bits/word)	
ROM Pack	JAMSC-MM40	EPROM, ROM Pack	
	JAMSC-MM41	EEPROM, ROM Pack	
Main Power Supply Module	JRMSP-PS40	For CPU, function modules and I/O modules (Max. 8)	Mounted on MB40
Auxiliary Power Supply Module	JRMSP-PS21	For I/O buffer modules and I/O modules (Max. 9)	Mounted on MB22
COMM Module	JAMSC-IF61	For communication module, P150 and MEMOBUS	2 ports/module
COMM Module with RAP I/F	JAMSC-IF41A	For communication module P150, MEMOBUS and RAP	2 ports/module
Servo I/F Module	JAMSC-IF66	Driving software servo up to 8.	
PC-LINK Module	JAMSC-IF64	Link between CPUs (Max. 32)	
Register Access Panel (RAP)	DISCT-IF69	Monitor modules for status and parameter	Connected to IF41A
I/O buffer Module	JAMSC-B2110A	Used for I/O bus buffer, and racks 2 to 4	With I/O cable connector
MB40 Mounting Base	JAMSI-MB40	For mounting CPU, power supply, COMM, Servo I/F and I/O modules	For rack 1
MB22 Mounting Base	JAMSI-MB22	For mounting I/O buffer, auxiliary power supply and I/O modules (Max. 9)	For racks 2 to 4
MB22S5 Mounting Base	JAMSI-MB22S5	For mounting I/O buffer, auxiliary power supply and I/O modules (Max. 5)	For racks 2 to 4
MB22S3 Mounting Base	JAMSI-MB22S3	For mounting I/O buffer, auxiliary power supply and I/O modules (Max. 3)	For racks 2 to 4
I/O Cables	JZMSZ-W20-1	Connection between racks 0.5m long	
	JZMSZ-W20-2	Connection between racks 1.5m long	
I/O Modules	JAMSC-B2501	100VAC 16-point input, input current 10mA/100VAC, 60Hz	
	JAMSC-B2503	200VAC 16-point input, input current 10mA/200VAC, 60Hz	
	JAMSC-B2505	100VAC 32-point input, input current 10mA/100VAC, 60Hz	
	JAMSC-B2507	200VAC 32-point input, input current 10mA/200VAC, 60Hz	
	JAMSC-B2601	12/24VDC 16-point input, input current 10mA/24VDC, 5mA/20VDC	

Component	Type	Function or Application	Remarks
I/O Modules	JAMSC-B2603	12/24VDC, 32-point input, input current 10mA/24VDC, 5mA/12VDC	
	JAMSC-B2605	12/24VDC 64-point input	
	JAMSC-B2607	5VDC 32-point input	
	JAMSC-B2500	100/200VAC 16-point output, rated output current 1A/circuit, 3A/8 circuits	
	JAMSC-B2504	100/200VAC 32-point output, rated output current 0.3A/circuit, 1.2A/8 circuits	
	JAMSC-B2600	12/24VDC 16-point output, rated output current 2A/circuit, 5A/8 circuits	With status indicator
	JAMSC-B2602	12/24VDC 32-point output, rated output current 0.3A/circuit, 0.6A/4 circuits	With status indicator
	JAMSC-B2604	12/24VDC 64-point output	
	JAMSC-B2606	5VDC 32-point output	
	JAMSC-B2902	Relay contact 32-point output, relay coil voltage 24VDC, rated current 1A/circuit (220VAC), 1A/circuit (24VDC)	
	JAMSC-B2904	Bestact relay contact 16-point output (independent contact), relay coil voltage 24VDC, small load rated current 0.5A/circuit (220VAC), 0.3A/circuit (110VDC), min. rating 5V, 1mA	
	JAMSC-B2914	Bestact relay contact 16-point output (independent contact), relay coil voltage 24VDC, general-use type rated current 0.5A/circuit (220VAC), 0.3A/circuit (110VDC), min. rating 24V, 10mA	
	JAMSC-B2701	Register input, numerical data (16-bit) ×8, strobe variable period (64/32ms)	
	JAMSC-B2700	Register output, numerical data (16-bit) ×8, strobe variable period (64/32ms)	
	JAMSC-B2703	Analog input (A/D) 0 to + 10V, 8 circuits	
	JAMSC-B2733	Analog input (A/D) −10 to + 10V, 8 circuits	

APPENDIX A

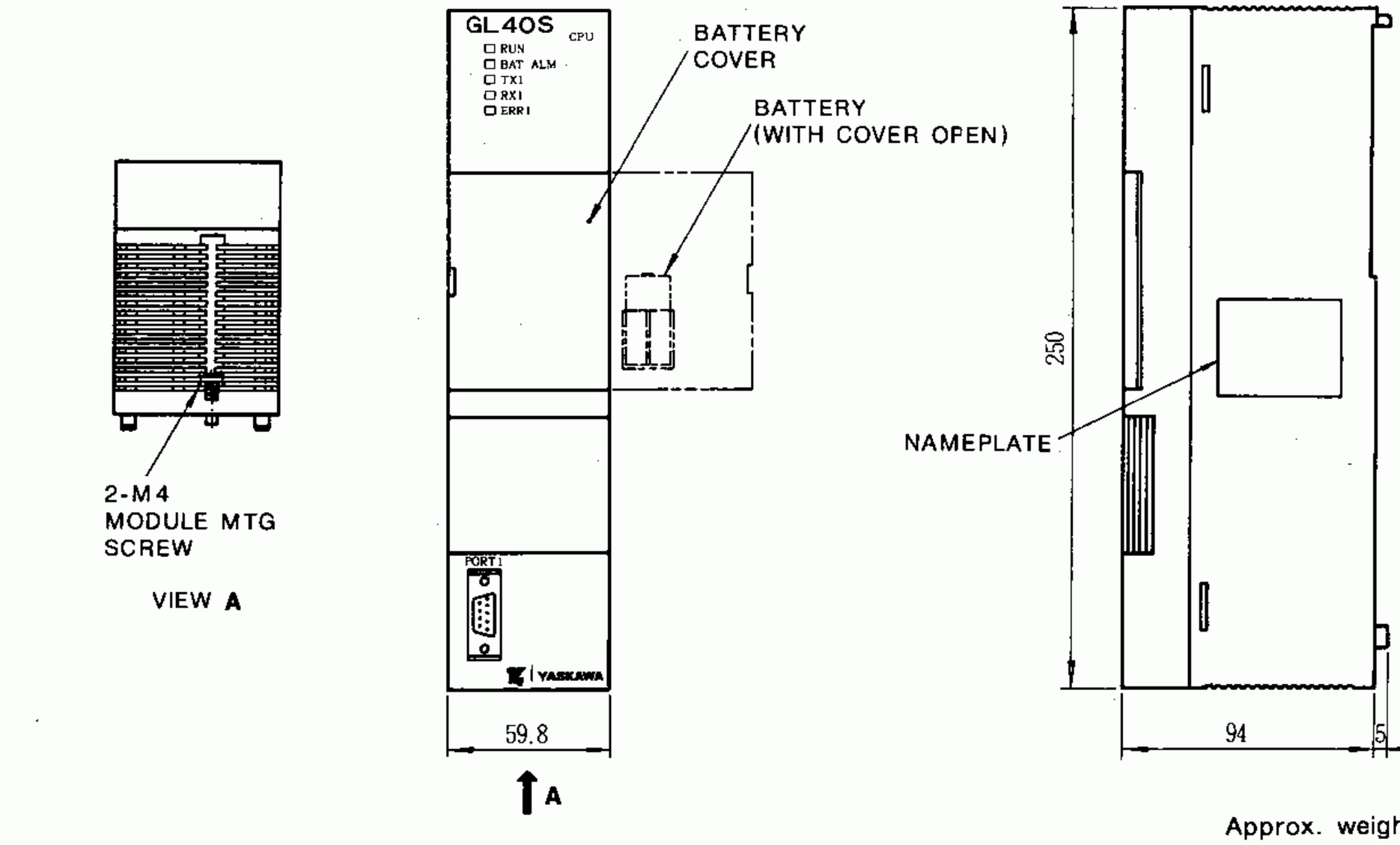
GL40S COMPONENTS LIST (Cont'd)

Component		Type	Function or Application	Remarks	
I/O Modules		JAMSC-B2743	Analog input (A/D) 40 to 20mA, 8 circuits		
		JAMSC-B2702	Analog output (D/A) 2 circuits		
		JAMSC-B2712	Analog output (D/A) 0 to +5V, 2 circuits		
		JAMSC-B2722	Analog output (D/A) −5 to + 5V, 2 circuits		
		JAMSC-B2732	Analog output (D/A) −10 to + 10V, 2 circuits		
		JAMSC-B2742	Analog output (D/A) 4 to 20mA, 2 circuits		
		JAMSC-B2801	Variable counter 2 circuits		
		JAMSC-B2802	Preset counter 1 circuit		
		JAMSC-B2803	Positioning (Speed analog command type) 1-axis corresponding to absolute encoder		
		JAMSC-B2813	Positioning (speed pulse command type) 1-axis		
		JAMSC-B2823	Positioning (speed pulse command type) 1-axis	For pulse motor	
		JAMSC-B2804	Memolink master		
		JAMSC-B2805	Memolink slave		
Programming Panel		DISCT-P150-10	Program storing, check, monitor, load, save	Program disk is needed	Full Key type
		DISCT-P150-11	Plasma display such as print-out, parameter setting		Sheet Key type
P150 Program Disk	Programmer	F40S-E001	For program storing,check, monitor, load or save		
	Ladder Lister	F40S-E002	For program print out		
	Blank Disk	F150-000	For program save	Format completed	
Interface Cable		JZMSZ-W1015-T1	For connection between P150-GL40S, COMM	2.5m long	
		JZMSZ-W1015-T2	For connection between module ports	15m long	
		JZMSZ-W1015-21	For connection between ACGC 400 series-GL40S, COMM module ports	2.5m long	
		JZMSZ-W1015-22		15m long	
		JZMSZ-W1017-T1	For connection between J1078 modem-GL40S, COMM module ports	5m long	
		JZMSZ-W1017-T2		15m long	
		JZMSZ-W1019-1	For connection between U84/U84S-GL40S	5m long	
		JZMSZ-W1019-2		15m long	

APPENDIX B

DIMENSIONS IN MM

(1) CPU Module (Type DDSCR-GL40S)

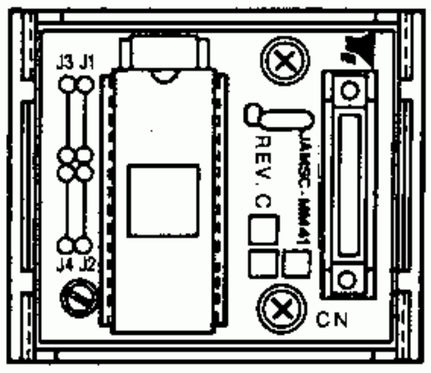


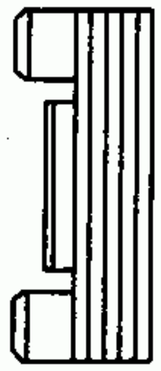
Approx. weight : 0.7kg

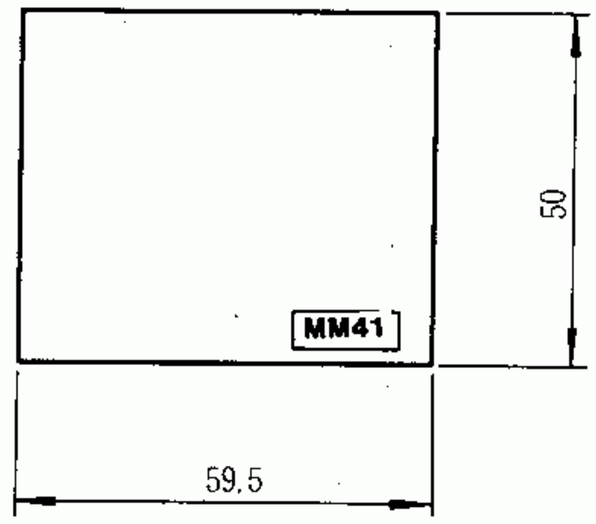
(2) EPROM Pack (Type JAMSC-MM40)

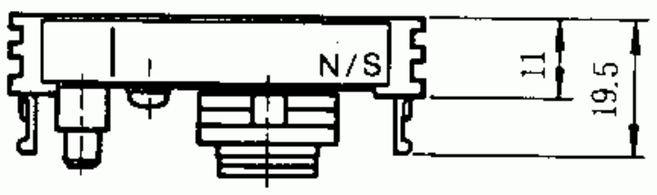
Approx. weight : 0.15kg

(3) EEPROM Pack (Type JAMSC-MM41)





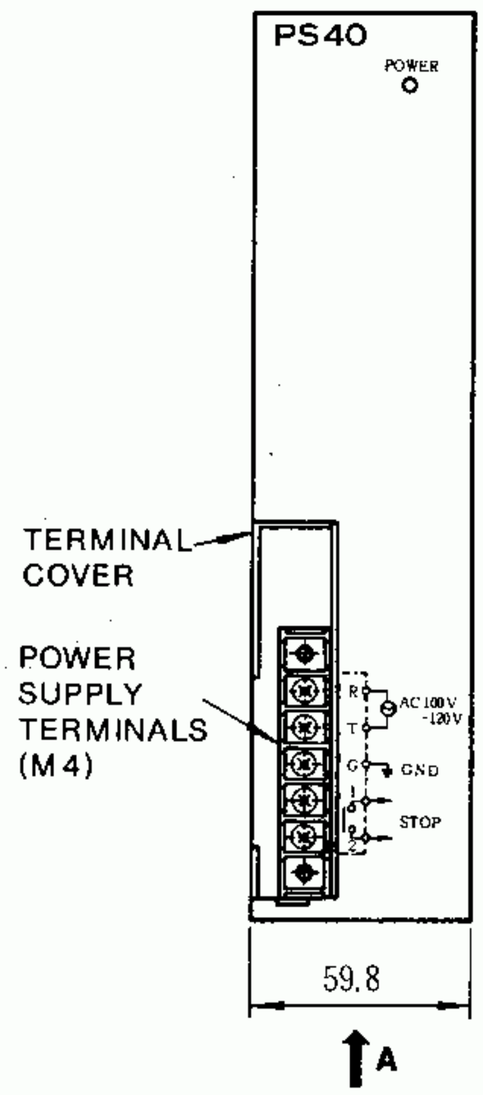


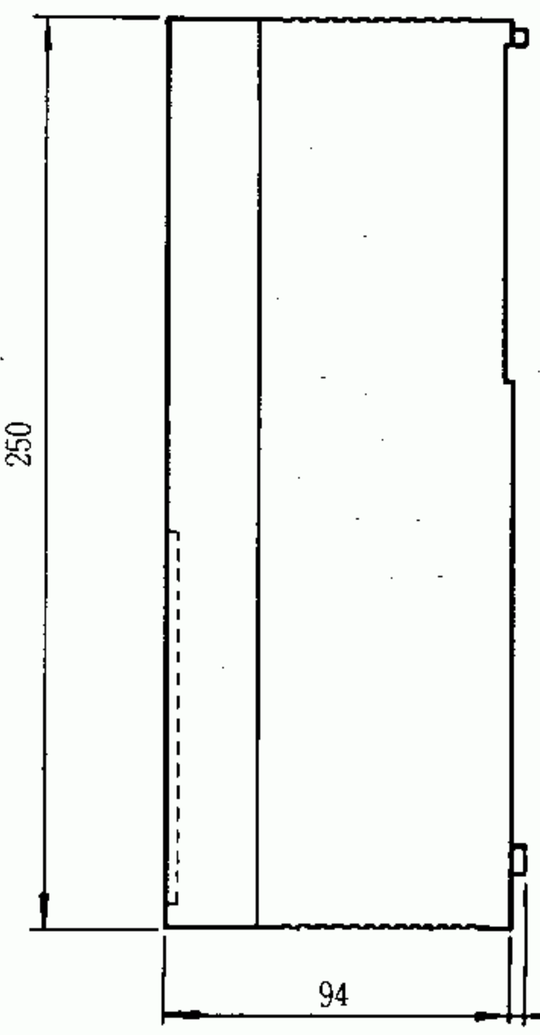


Approx. weight : 0.15kg

(4) Main Power Supply Module (Type JRMSP-PS40)

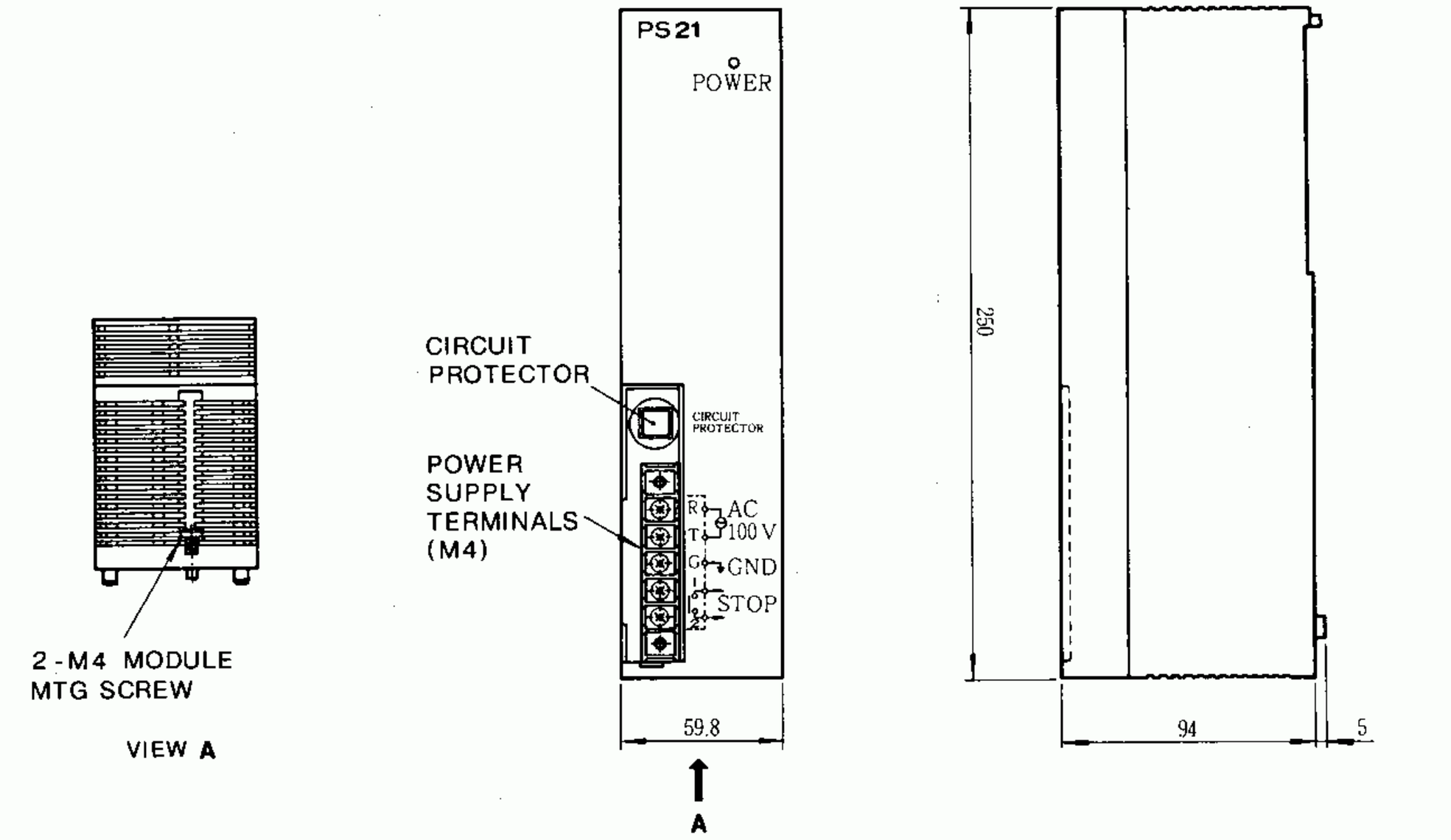
VIEW **A**





Approx. weight : 0.8kg

(5) Auxiliary Power Supply Module (Type JRMSP-PS21)

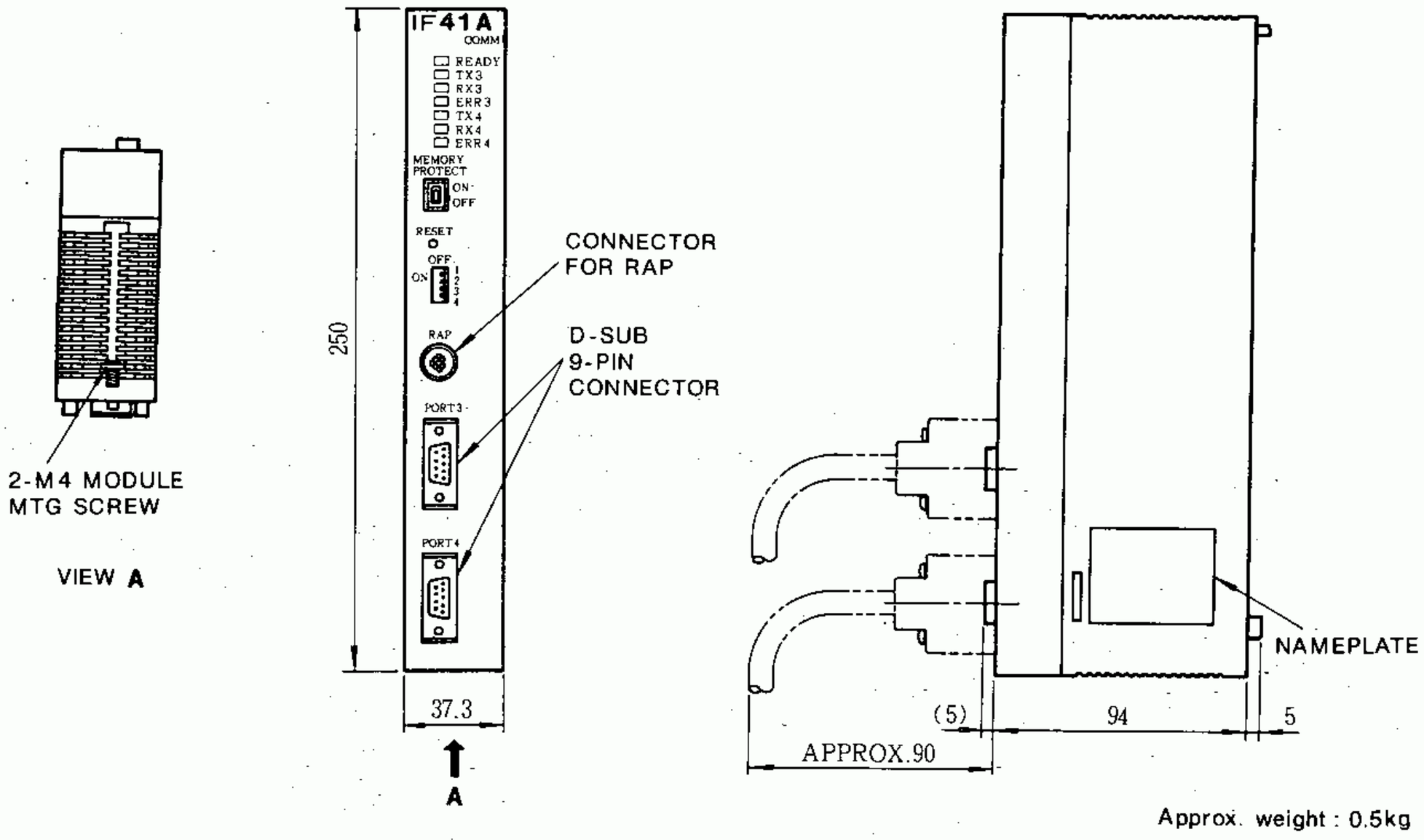


Approx. weight : 0.7 kg

(6) COMM Module (Type JAMSC-IF61)

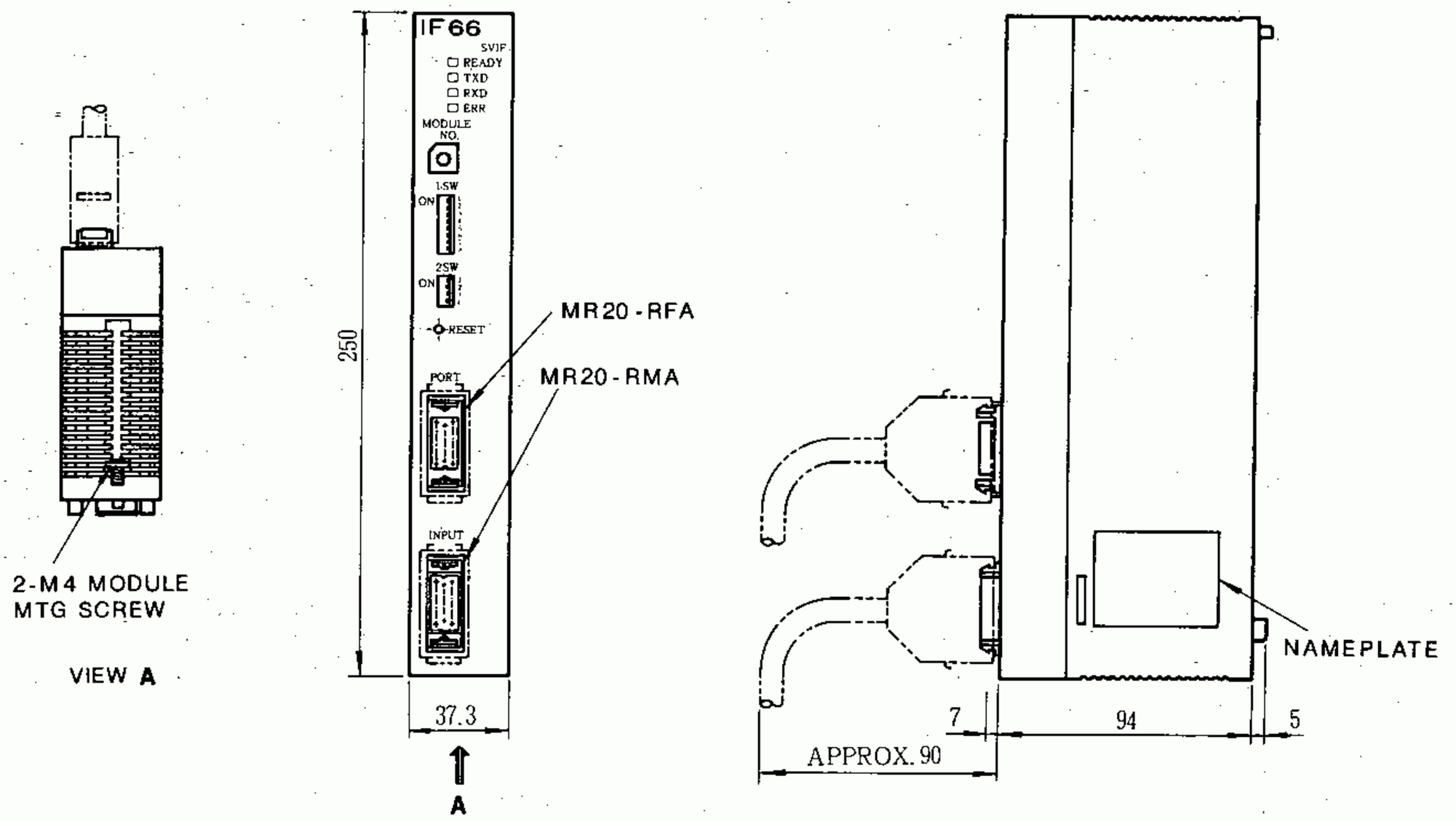
Approx. weight: 0.6 kg

(7) COMM Module with R-AP Port (Type JAMSC-IF41A)



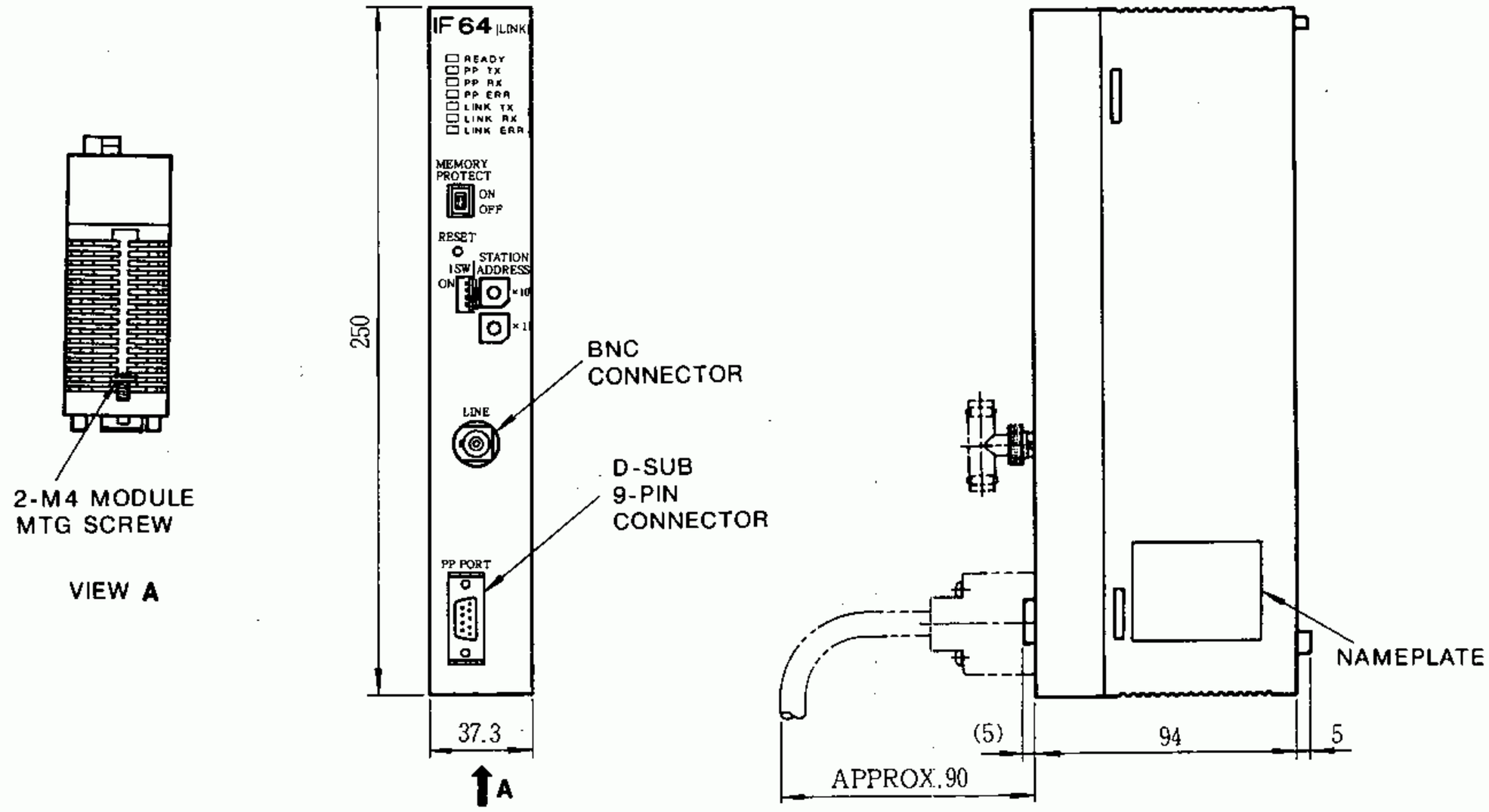
Approx. weight : 0.5kg

(8) SERVO I/F Module (Type JAMSC-IF66)



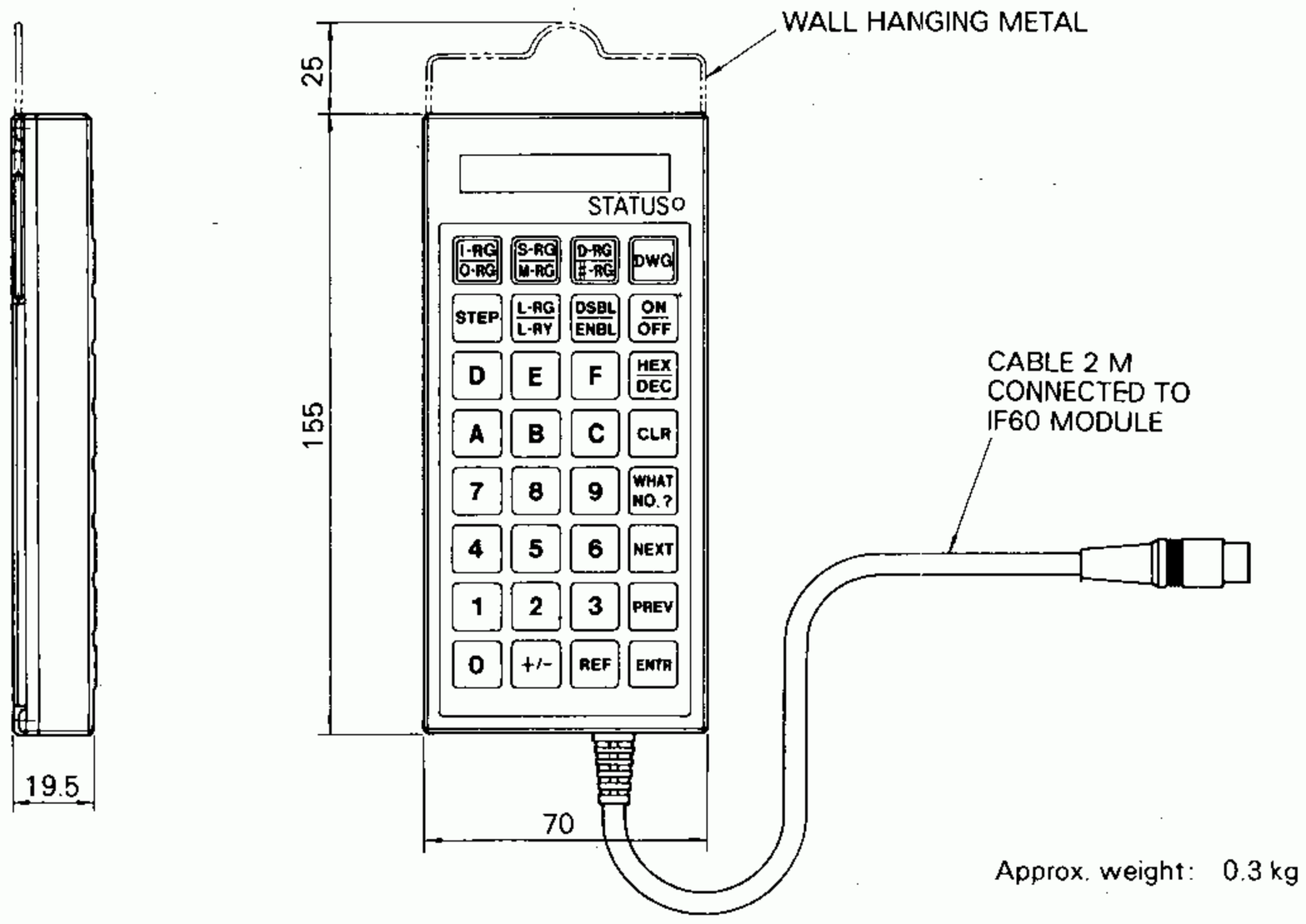
Approx. weight : 0.6kg

(9) PC-LINK Module (Type JAMSC-IF64)



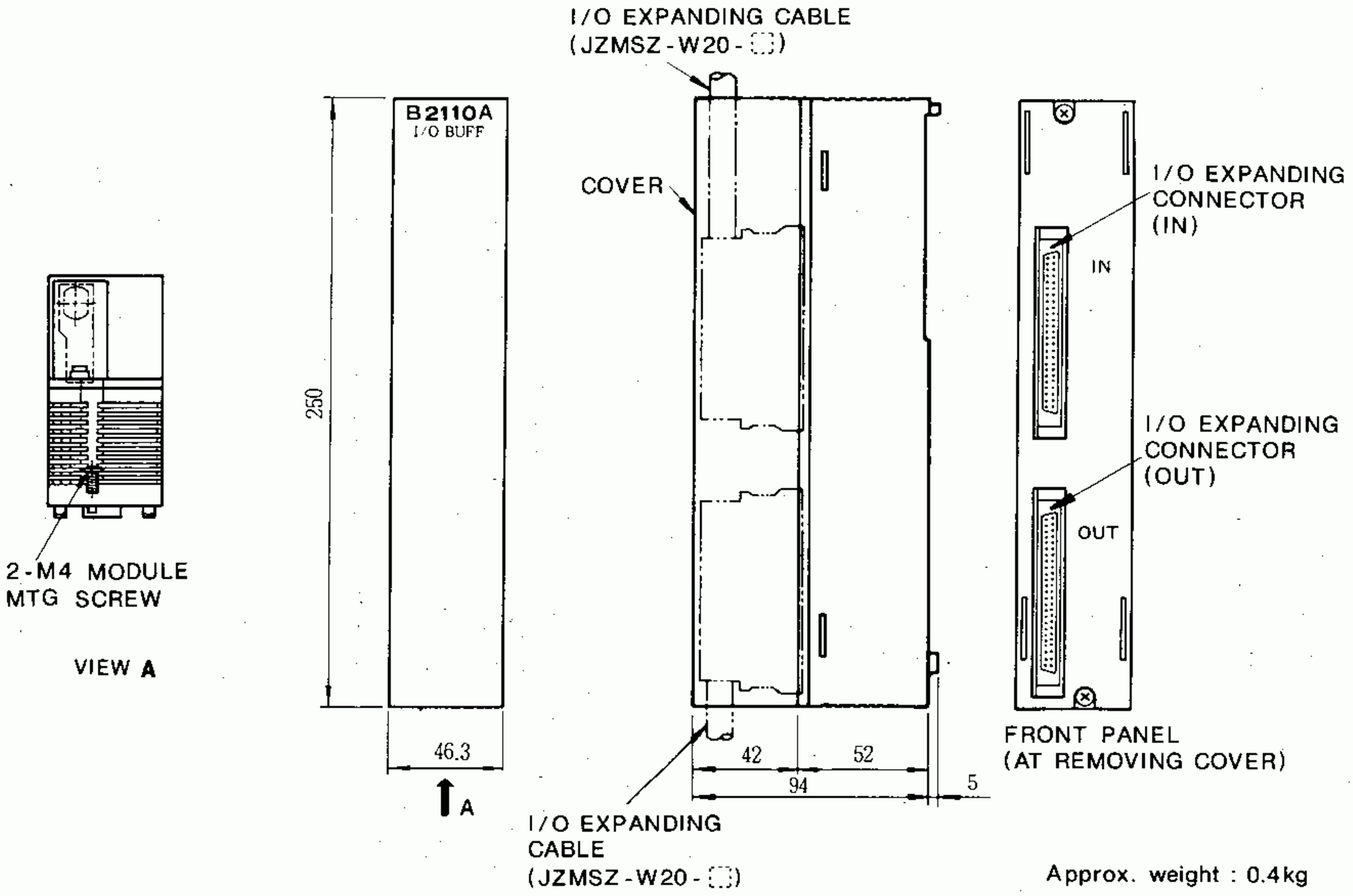
Approx. weight : 0.6kg

(10) Register Access Panel : RAP (Type DISCT-IF69)



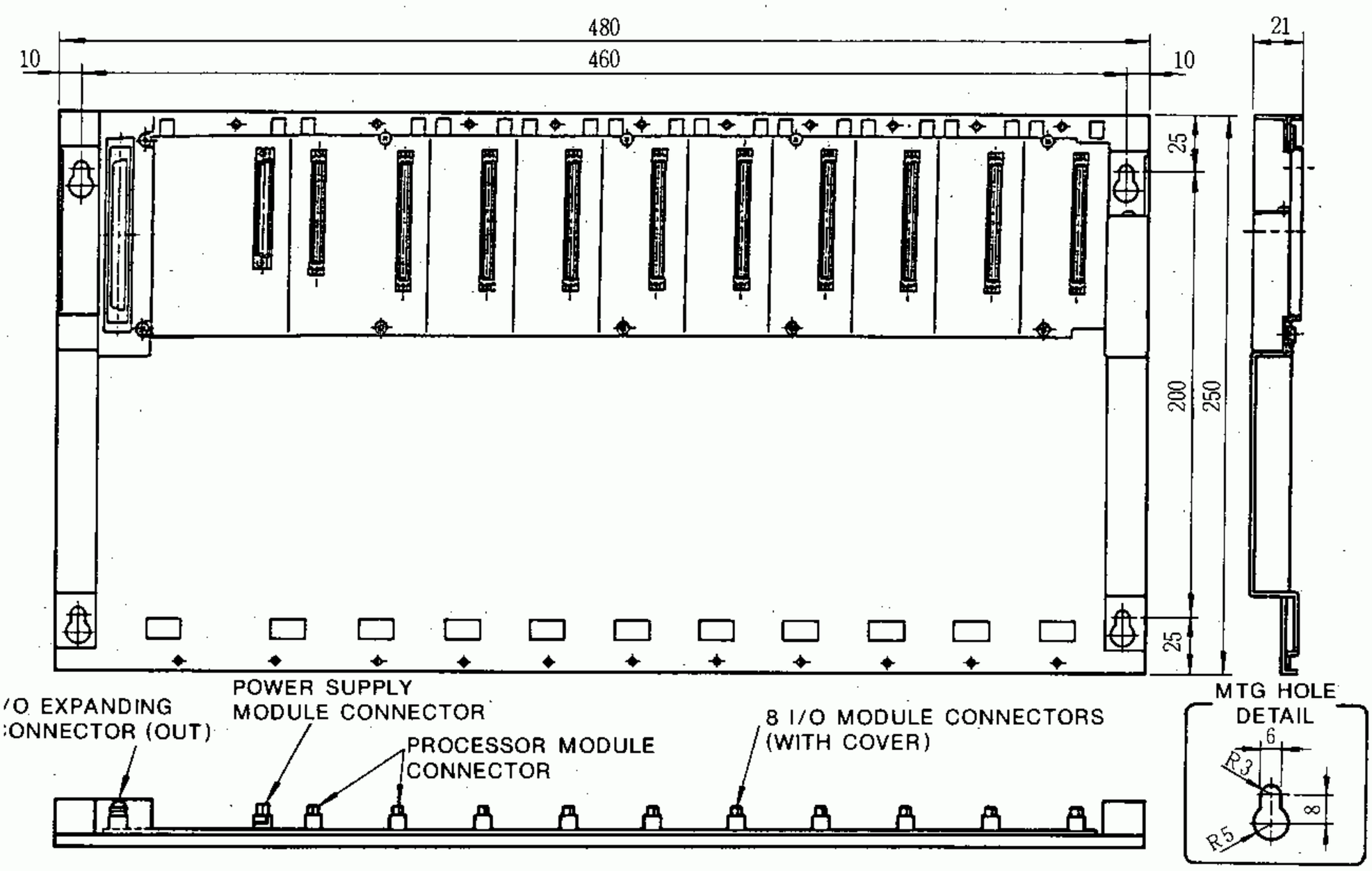
Approx. weight: 0.3 kg

(11) I/O Buffer Module (Type JAMSC-B2110A)



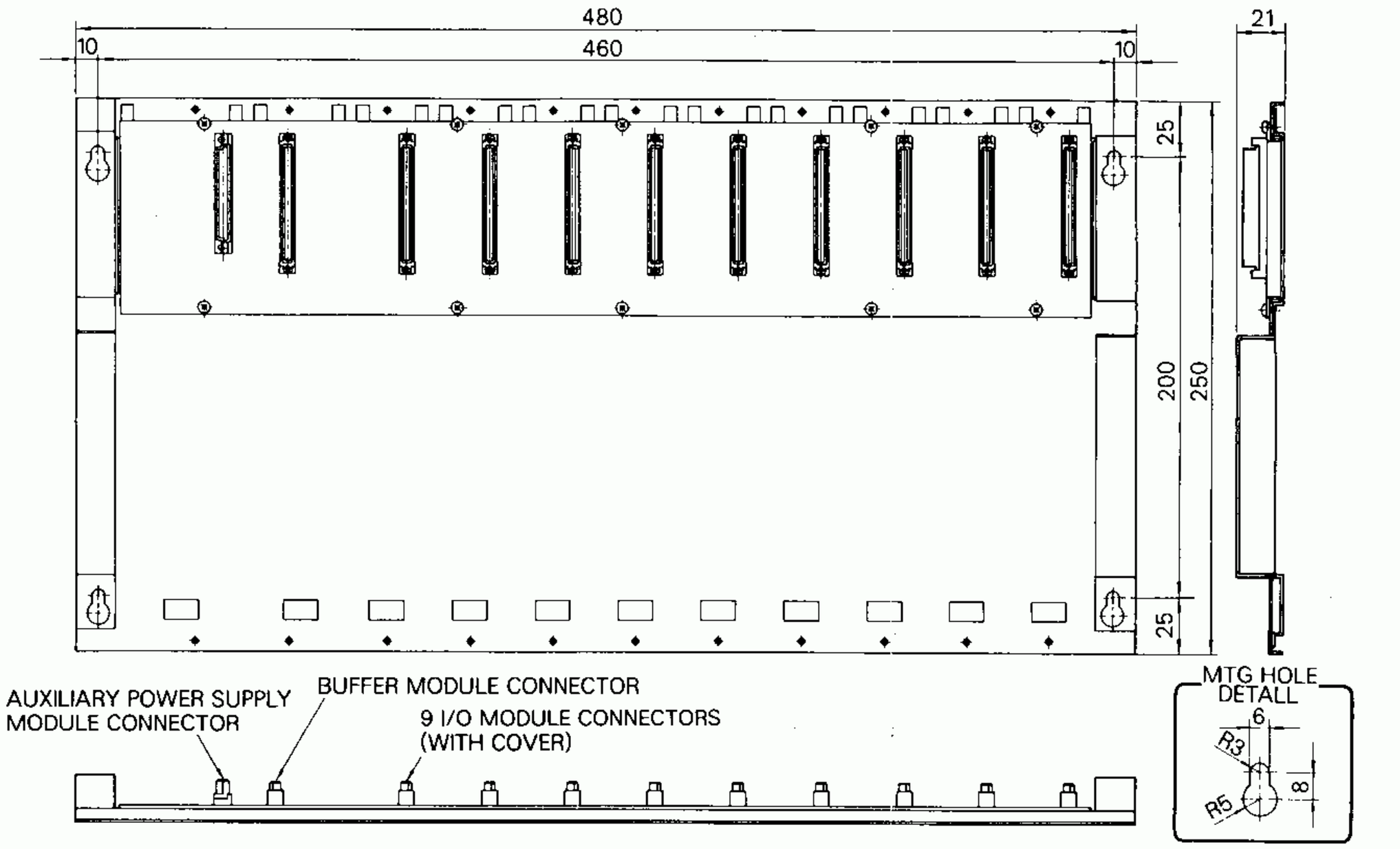
Approx. weight : 0.4kg

(12) MB40 Mounting Base (Type JRMSI-MB40)



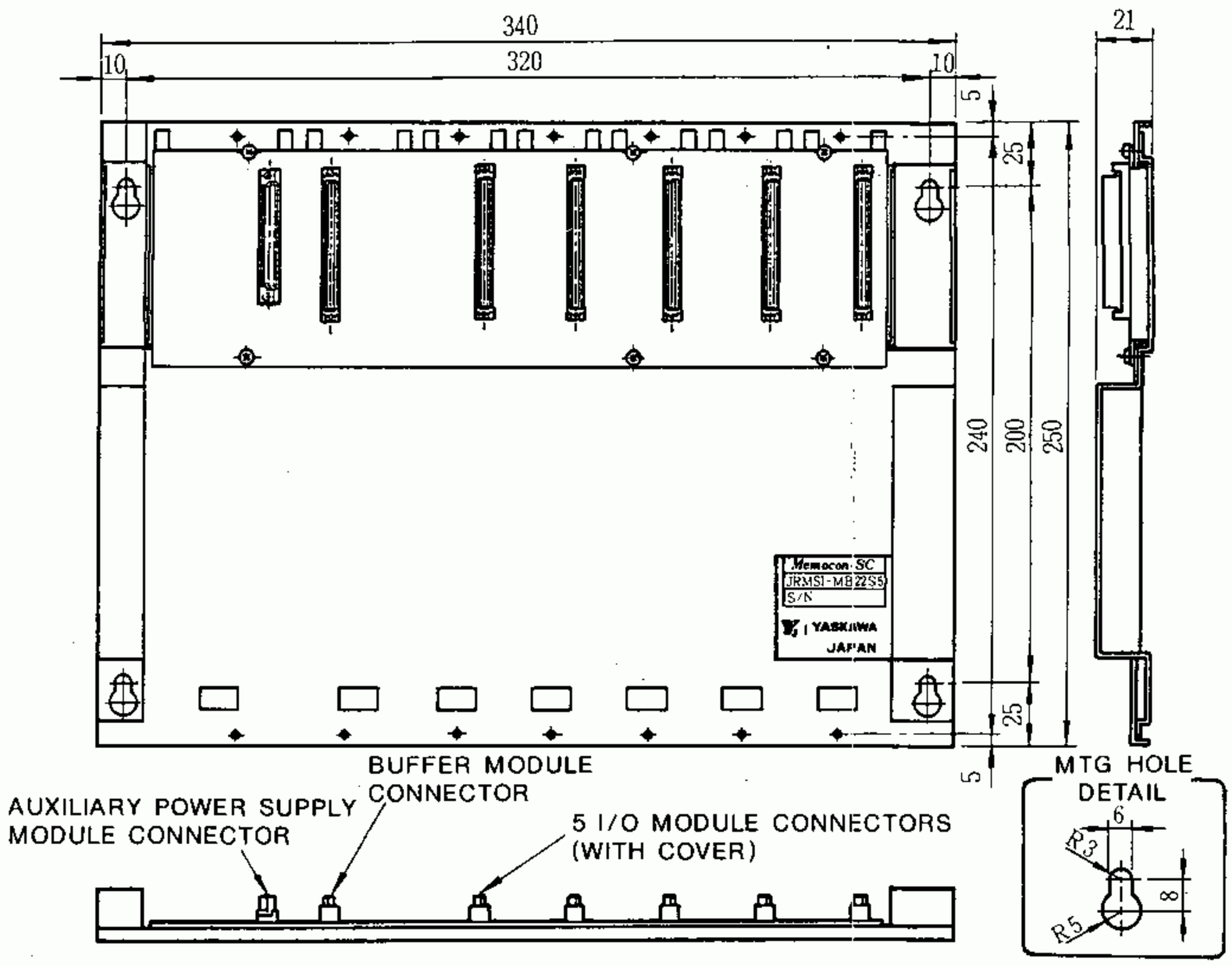
Approx. weight : 1.4kg

(13) MB22 Mounting Base (Type JRMSI-MB22)



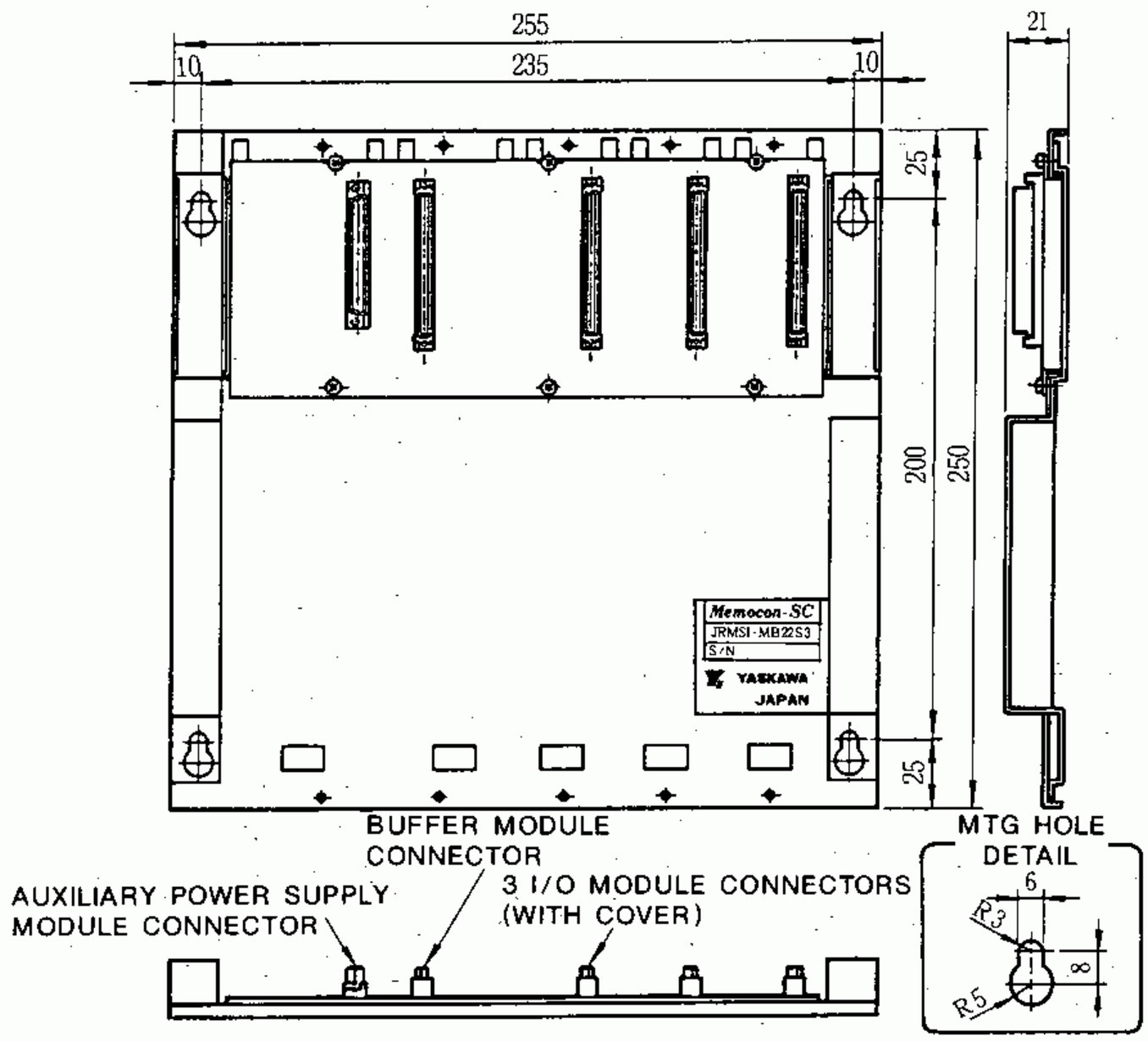
Approx. weight: 1.3 kg

(14) MB22S5 Mounting Base (Type JRMSI-MB22S5)



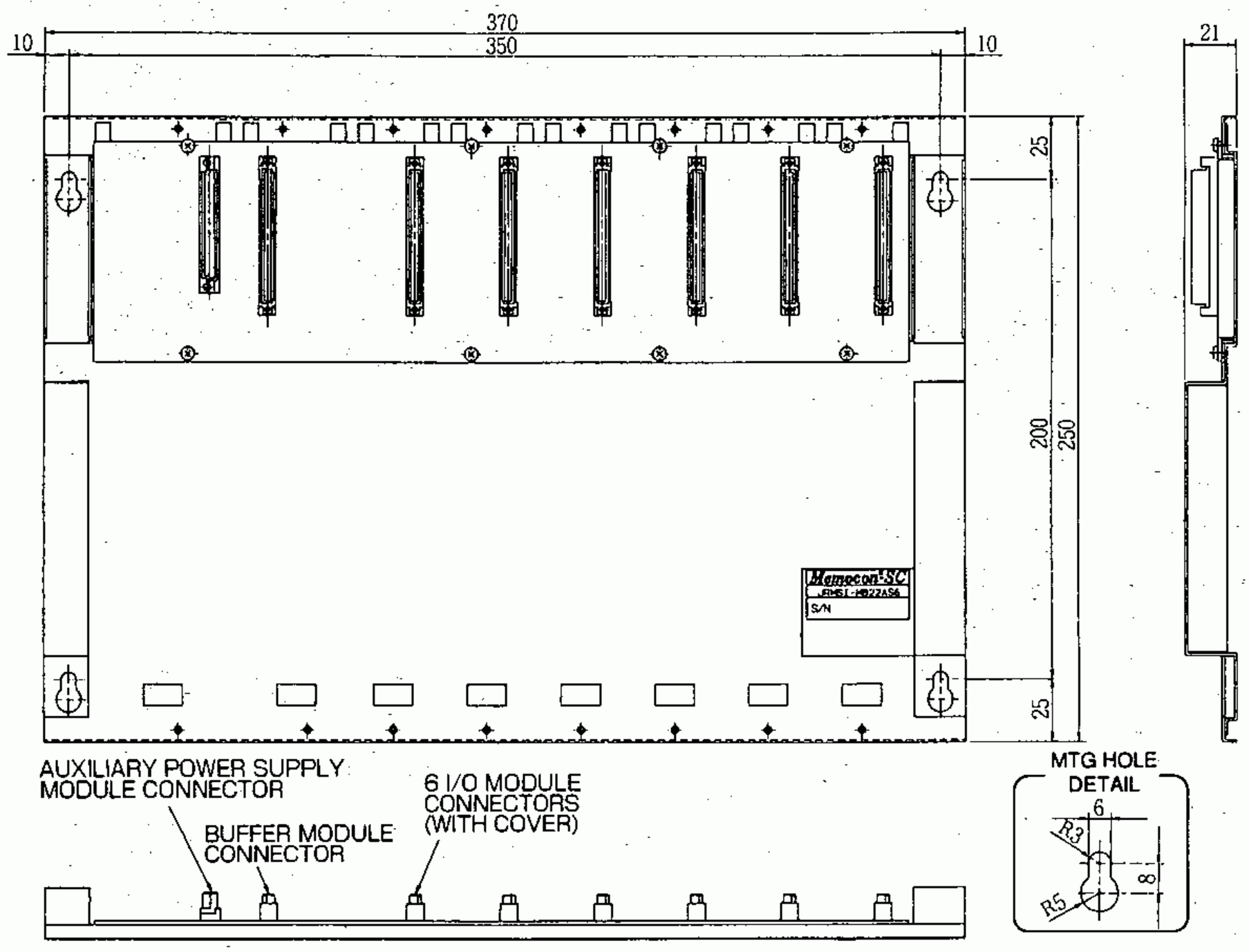
Approx. weight : 1kg

(15) MB22S3 Mounting Base (Type JRMSI-MB22S3)



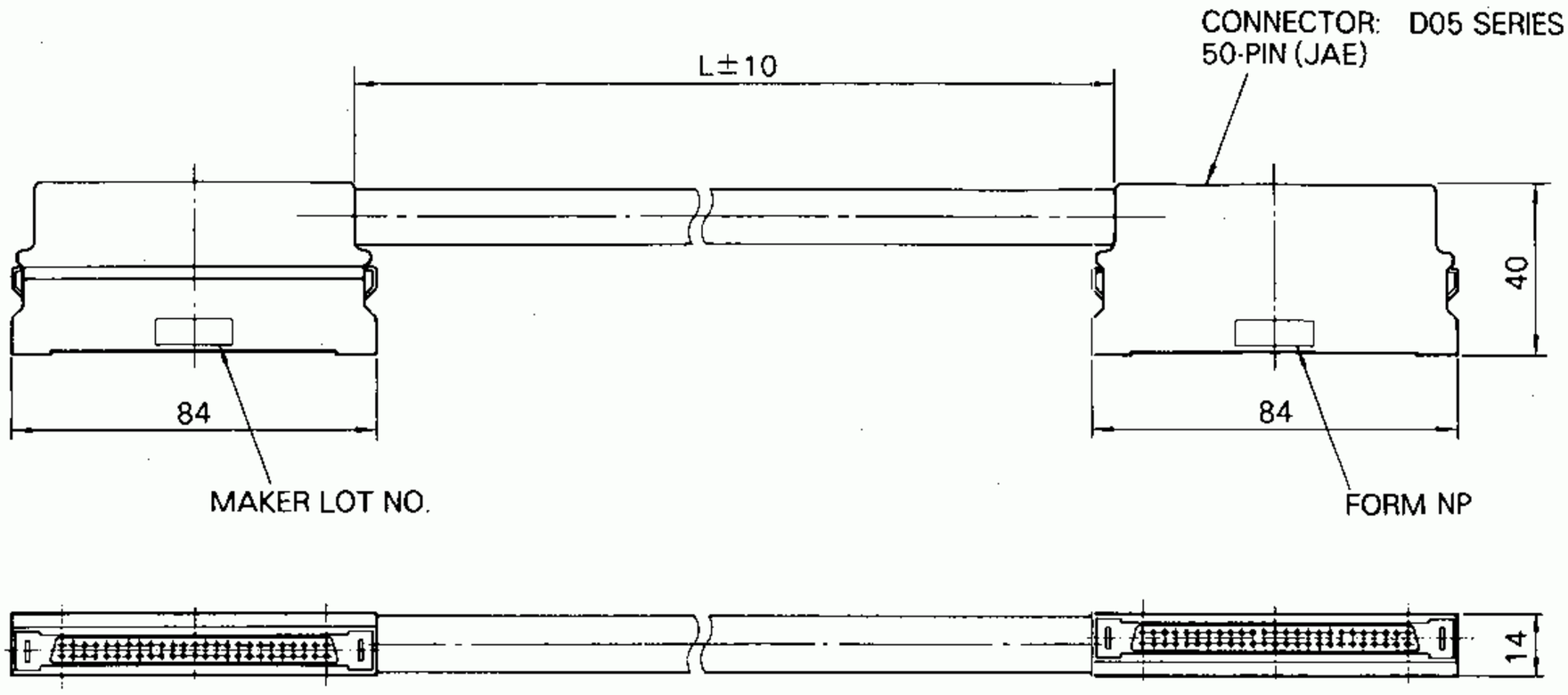
Approx. weight : 0.8kg

(16) MB22AS6 Mounting Base (Type JRMSI-MB22AS6)



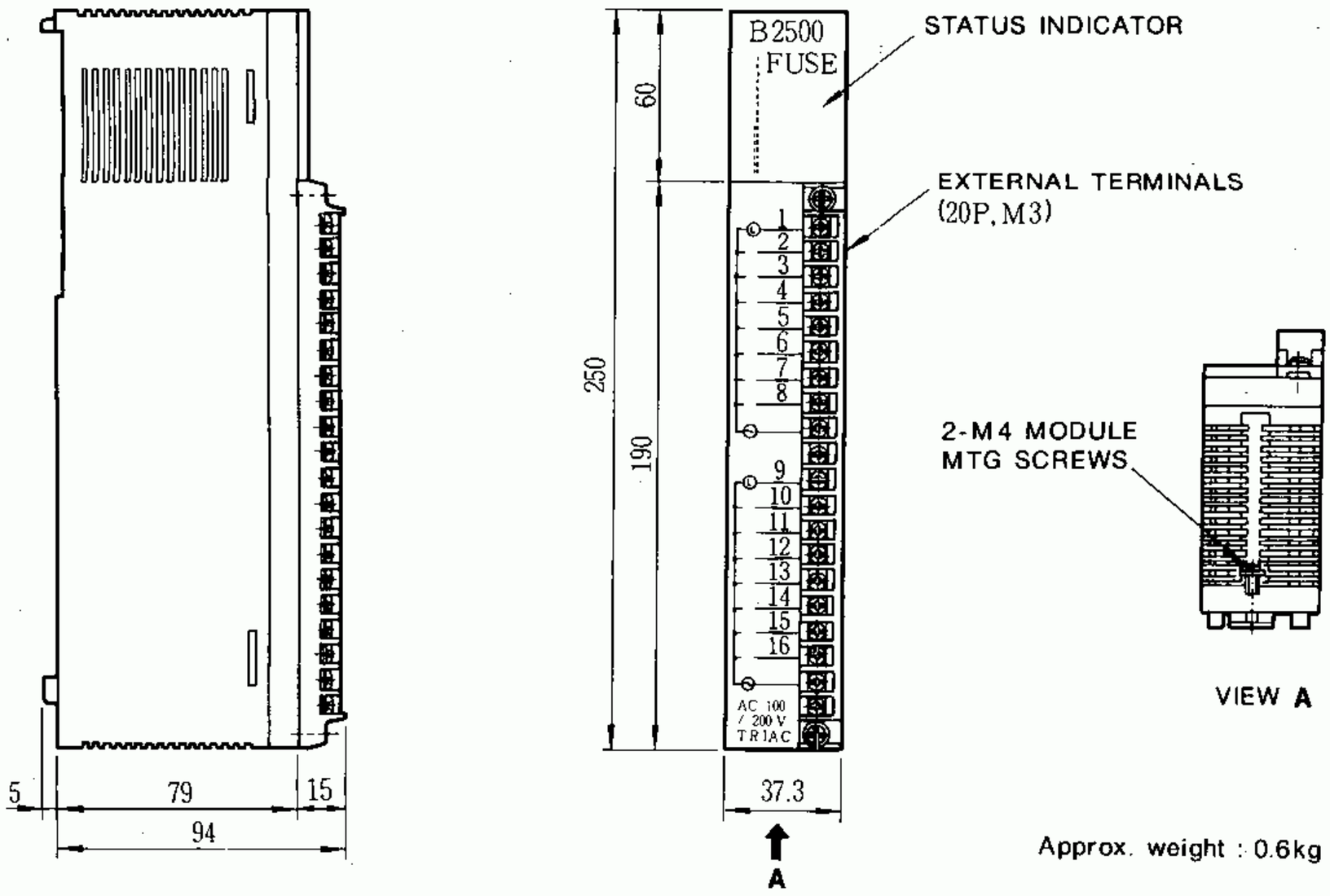
Approx. weight : 0.8 kg

(17) I/O Cable (Type JZMSZ-W20-1, -2)

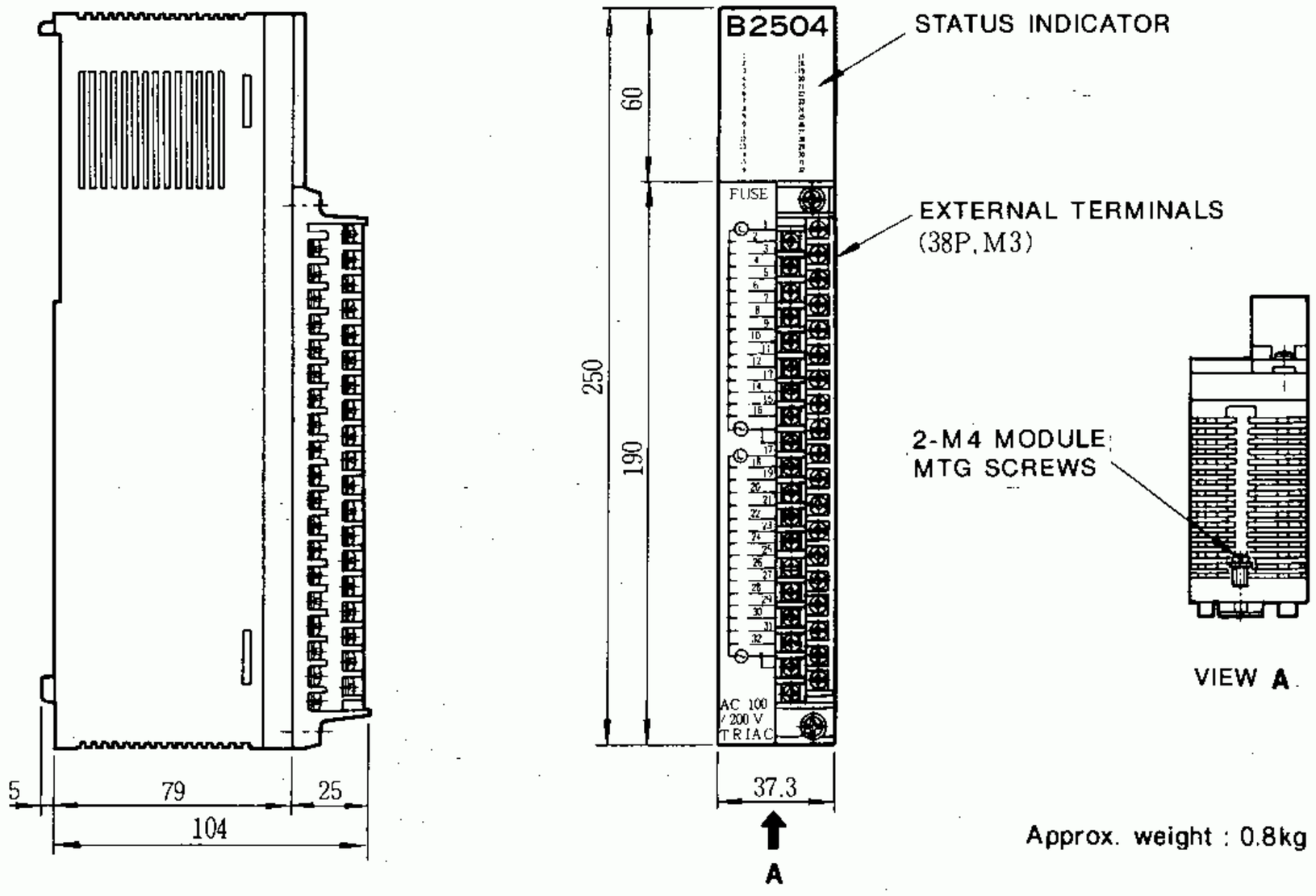


Type JZMSZ-	Length
W 20 - 1	0.5 m
W 20 - 2	1.5 m

(18) 16-I/Os Module (Types JAMSC-B2500, B2501, B2503, B2600, B2601, B2611, B2800, B2806)

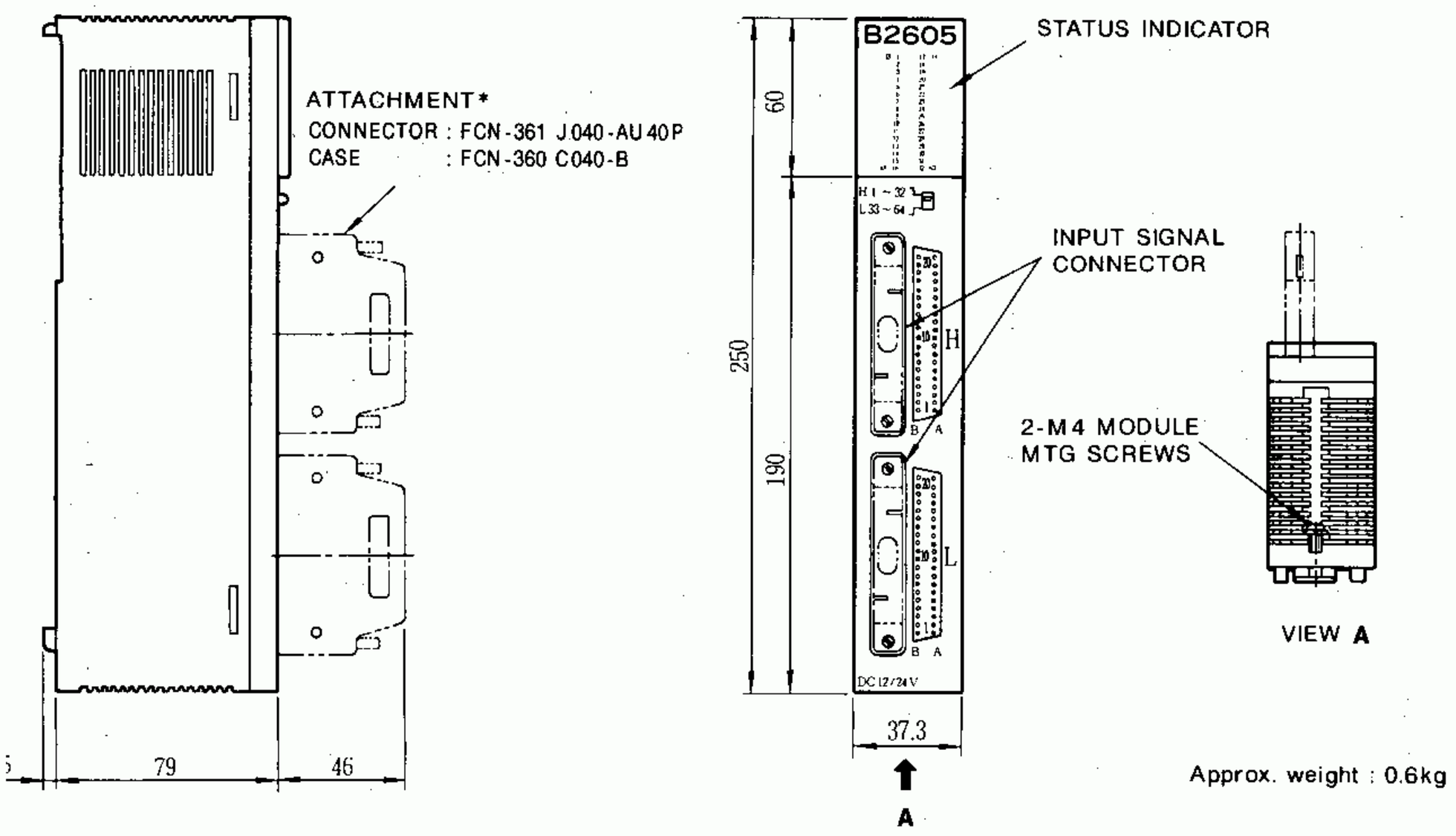


(19) 32-I/Os Module (Types JAMSC-B2504, B2505, B2507, B2602, B2603, B2606, B2607, B2700, B2701, B2801, B2802, B2902, B2904)



Approx. weight : 0.8kg

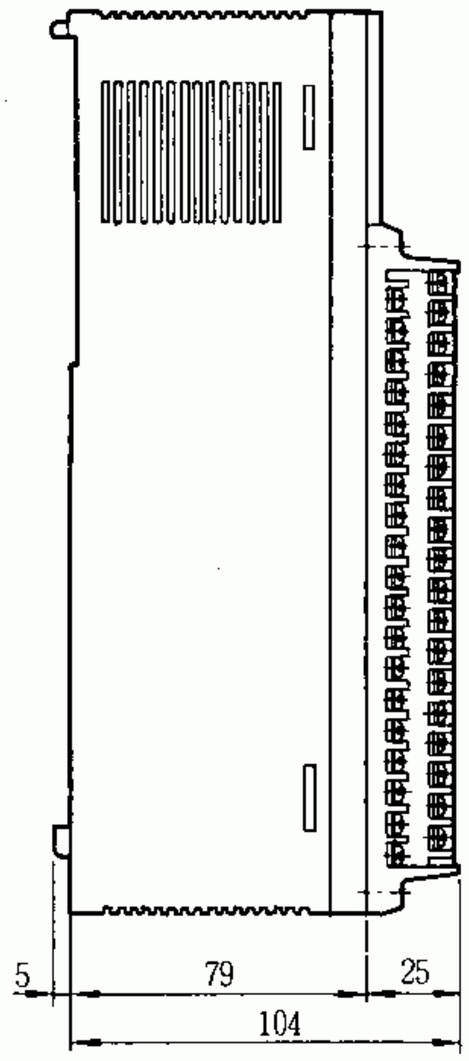
(20) 64-I/Os Module (Types JAMSC-B2605, B2604)

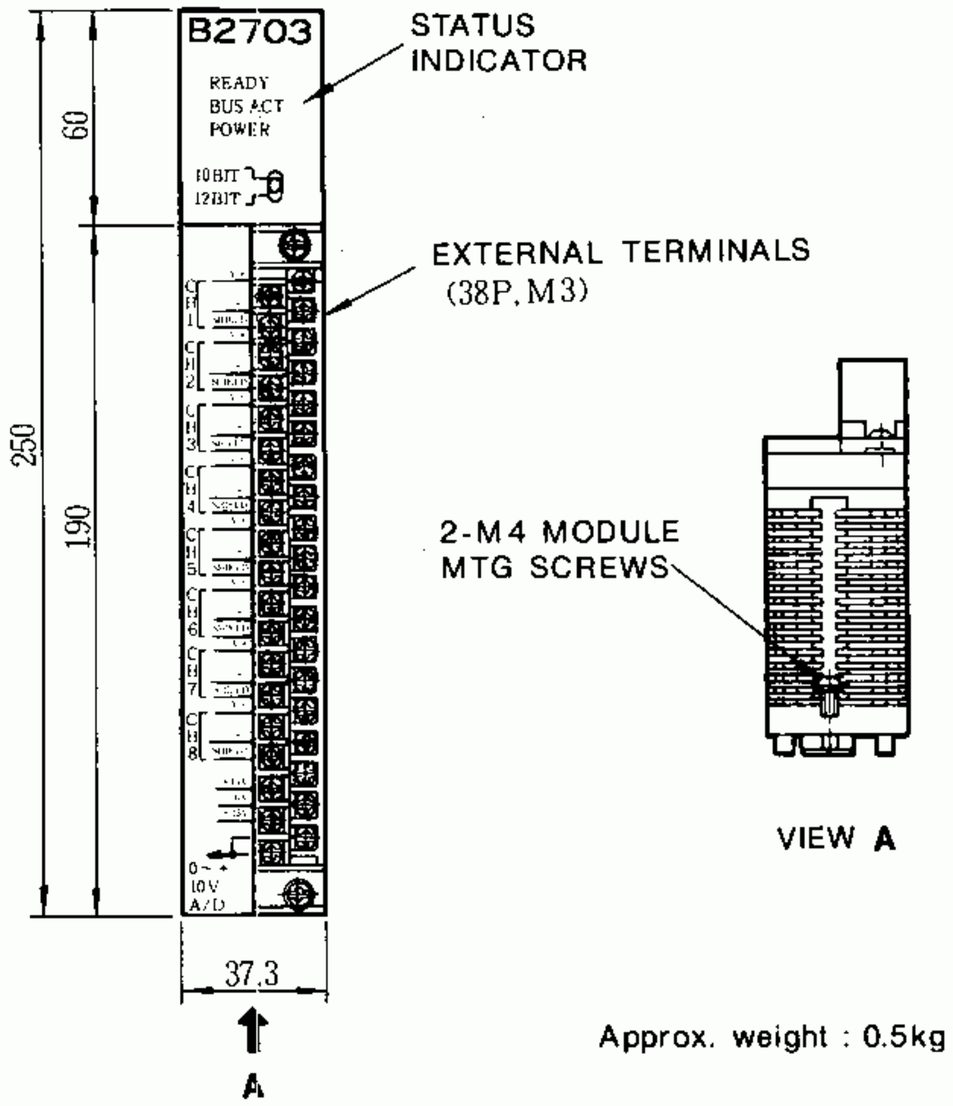


Approx. weight : 0.6kg

* Made by FUJITSU LTD.

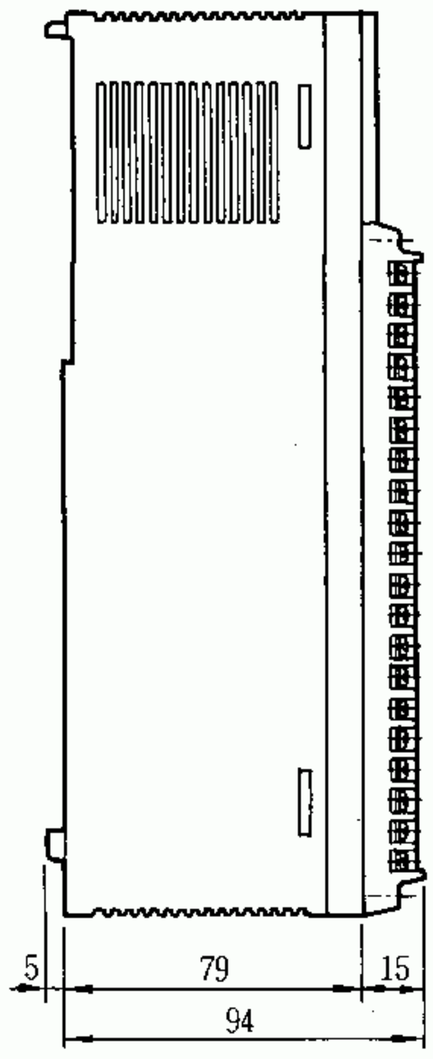
(21) Analog Input Module (Types JAMSC-B2703, B2733, B2743)

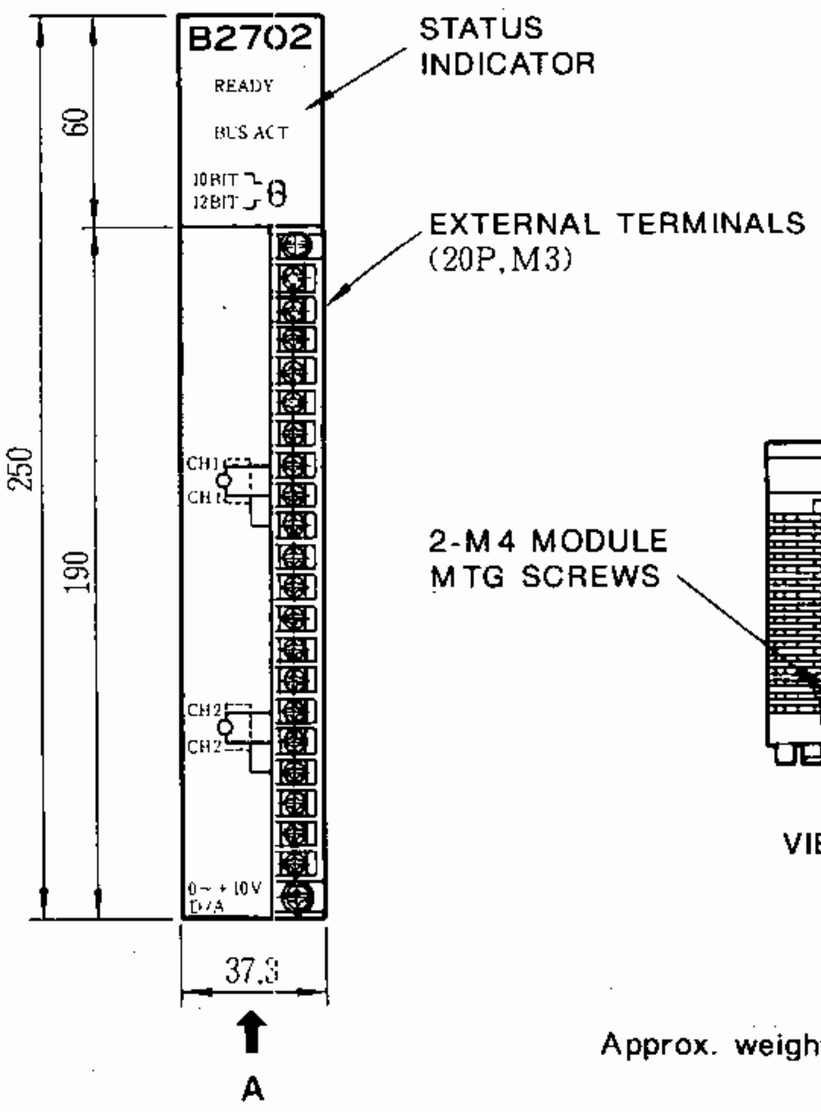




Approx. weight : 0.5kg

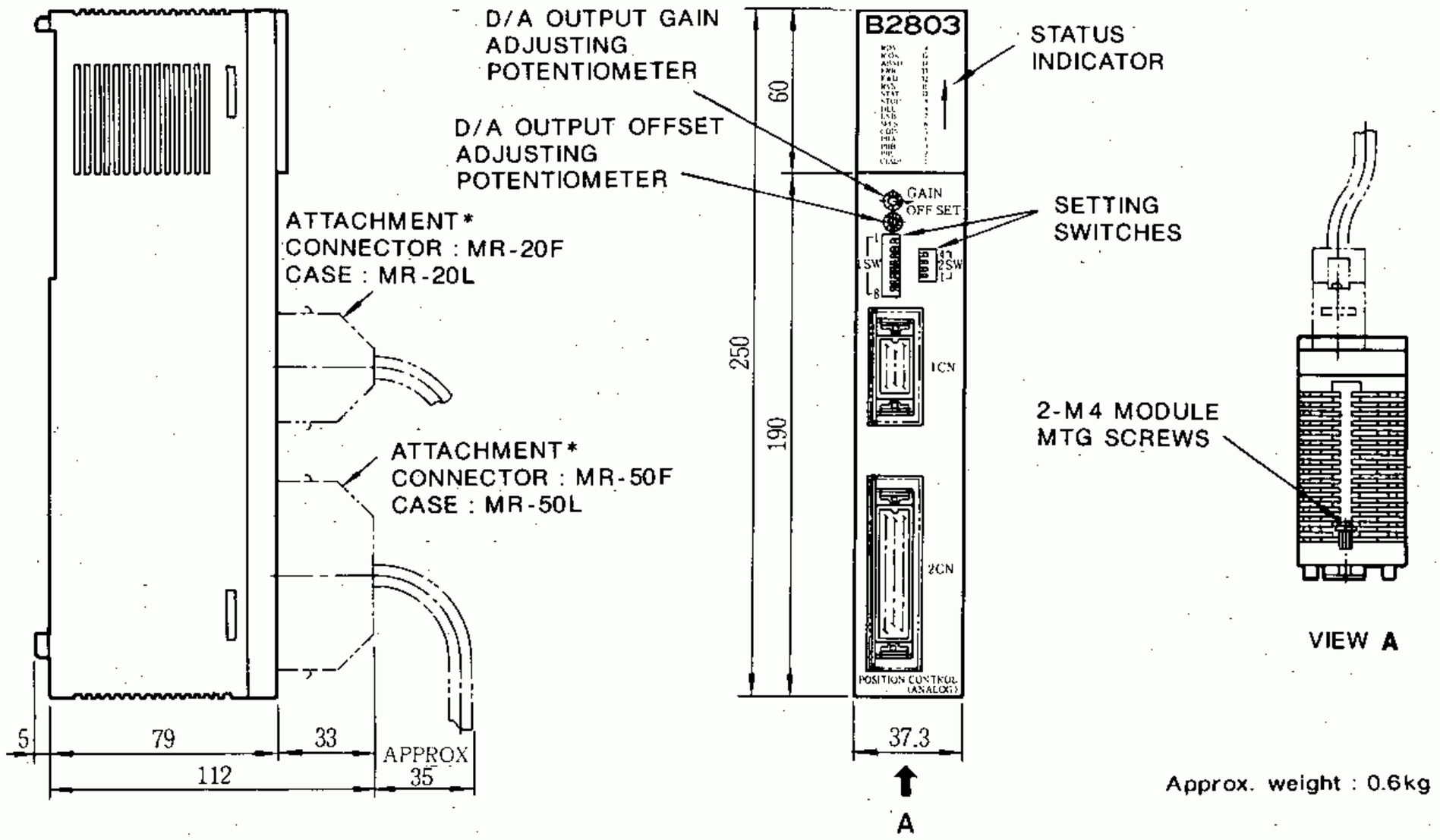
(22) Analog Output Module (Types JAMSC-B2702, B2712, B2722, B2732, B2742)





Approx. weight : 0.6kg

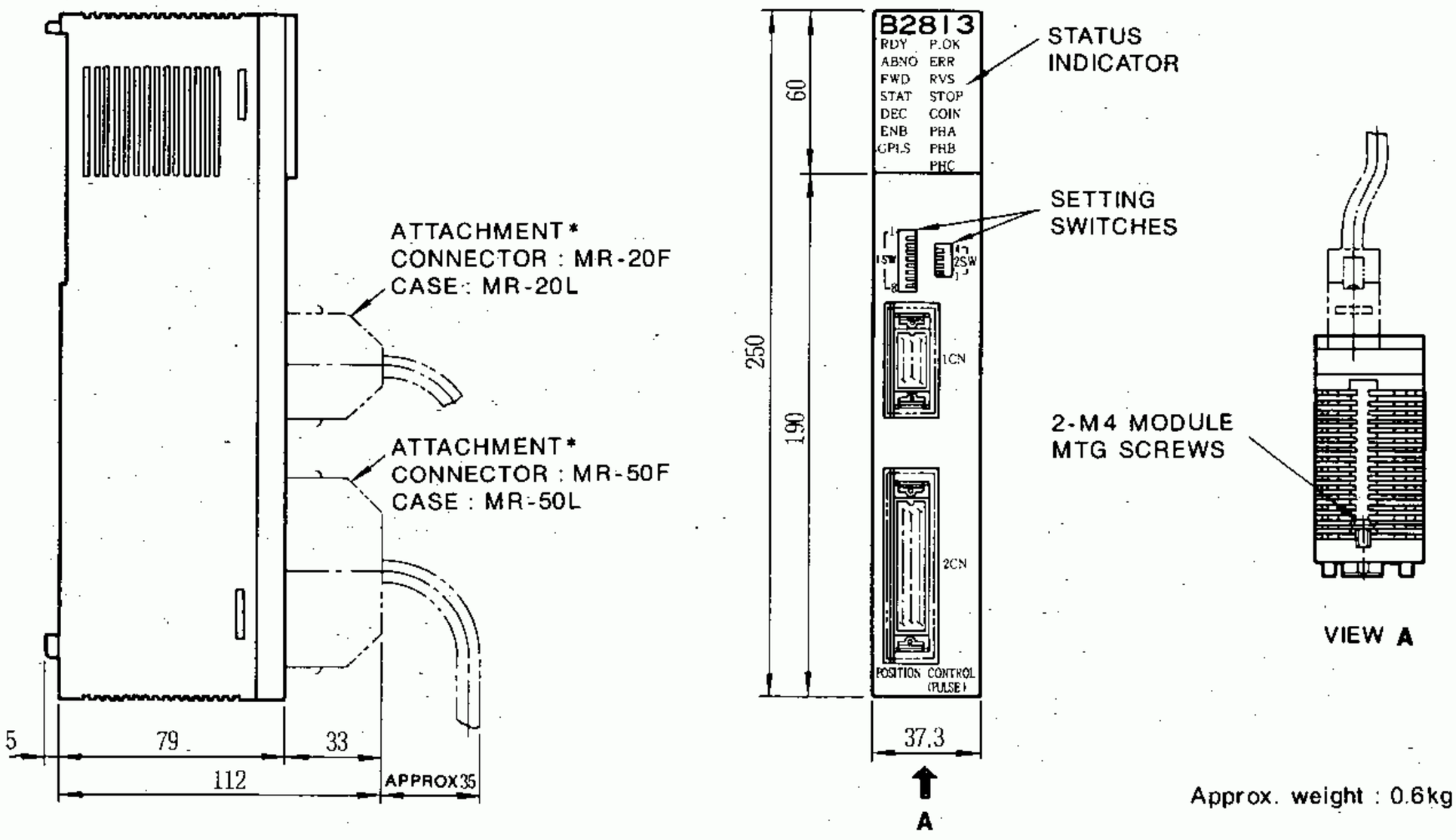
(23) Positioning Module (Types JAMSC-B2803)



Approx. weight : 0.6kg

* Made by Honda Tsushin Co.,Ltd.

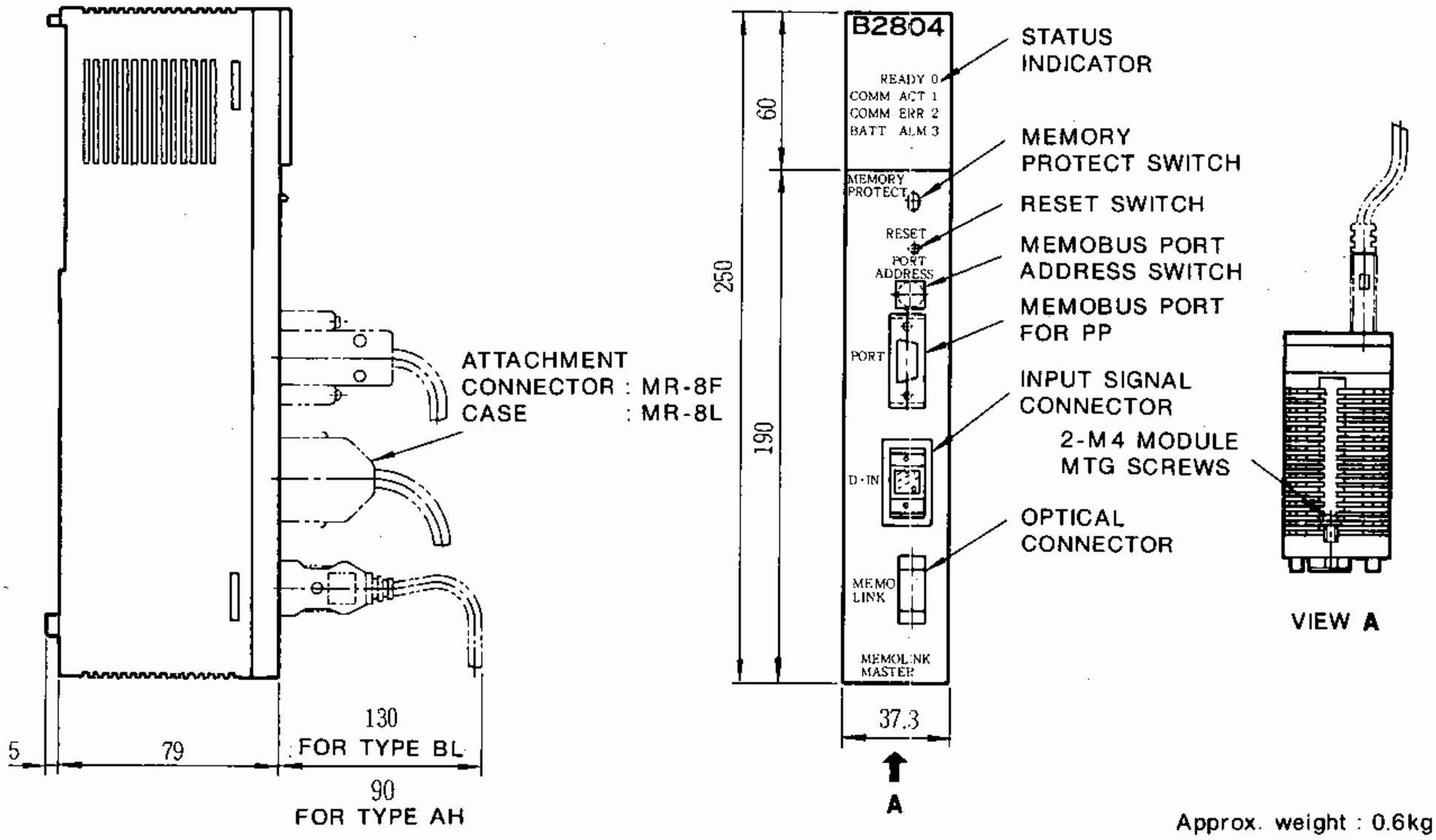
(24) Positioning Module (Types JAMSC-B2813, B2823)



Approx. weight : 0.6kg

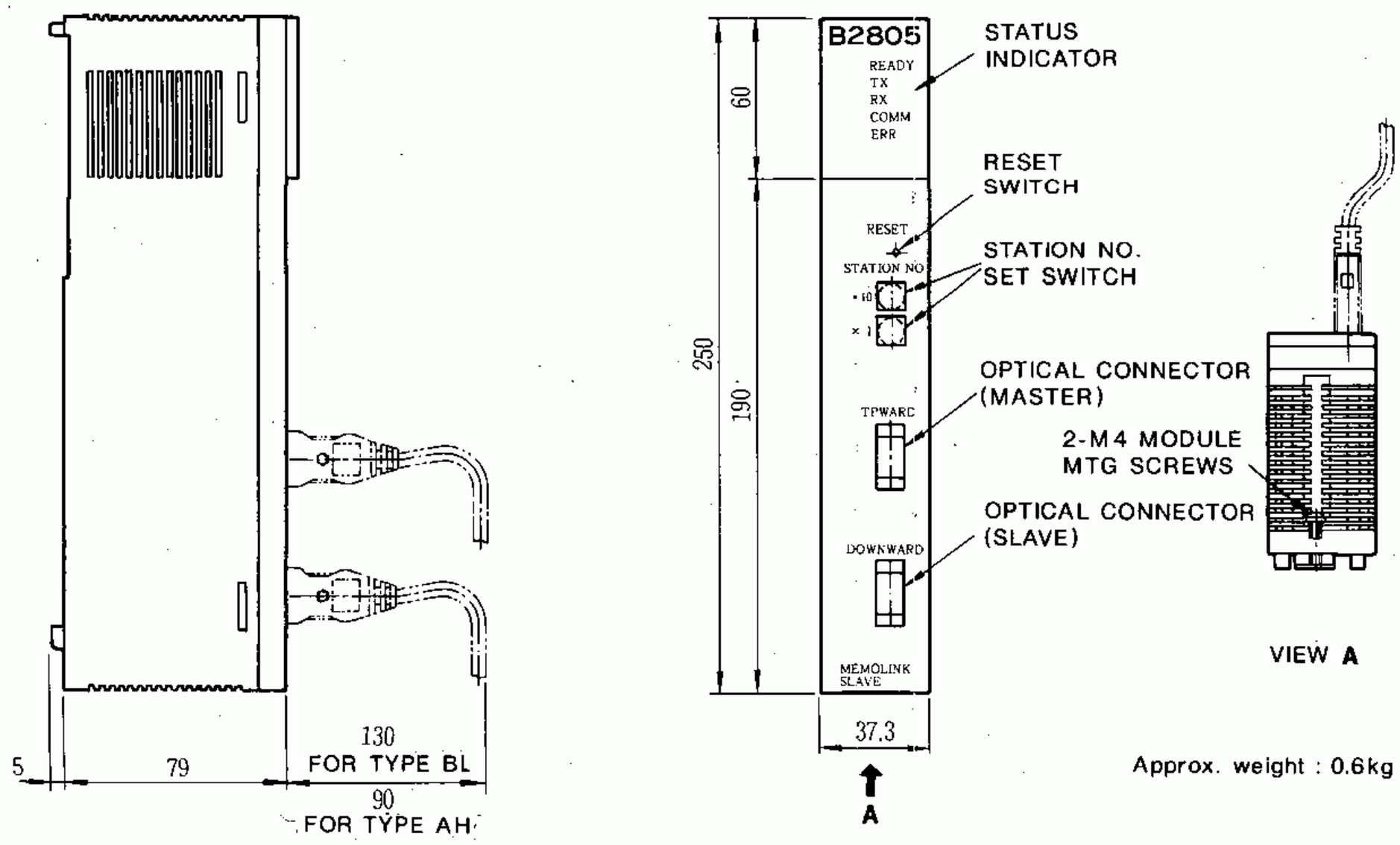
* Made by Honda Tsushin Co.,Ltd.

(25) MEMOLINK Master Module (Type JAMSC-B2804)



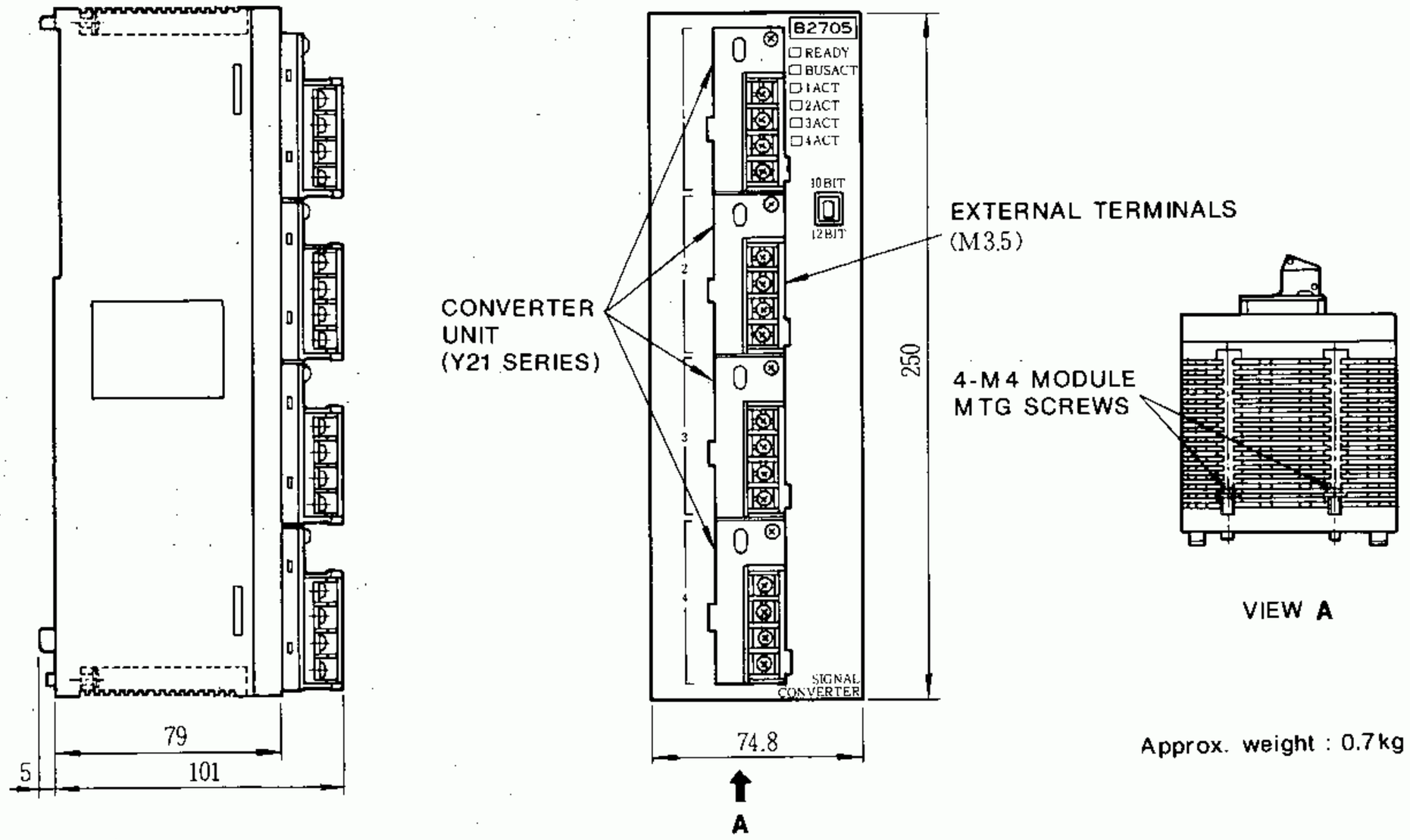
Approx. weight : 0.6kg

(26) MEMOLINK Slave Module (Type JAMSC-B2805)



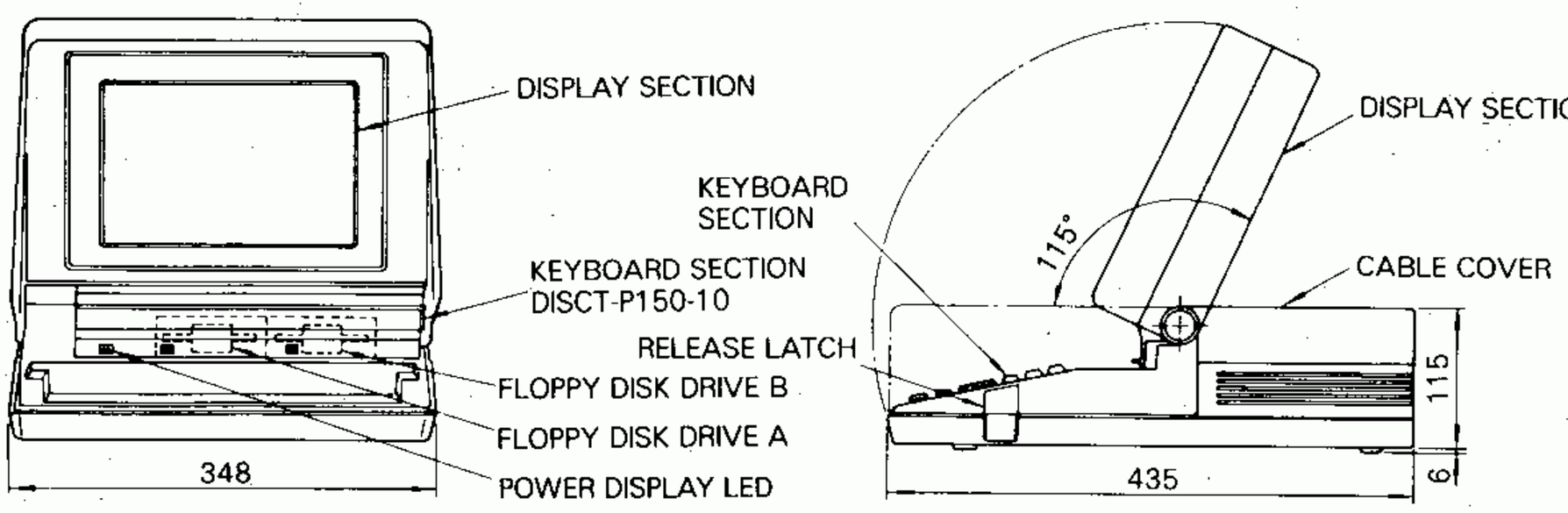
Approx. weight : 0.6kg

(27) Signal Converter Module (Type JAMS-B2705)



Approx. weight : 0.7kg

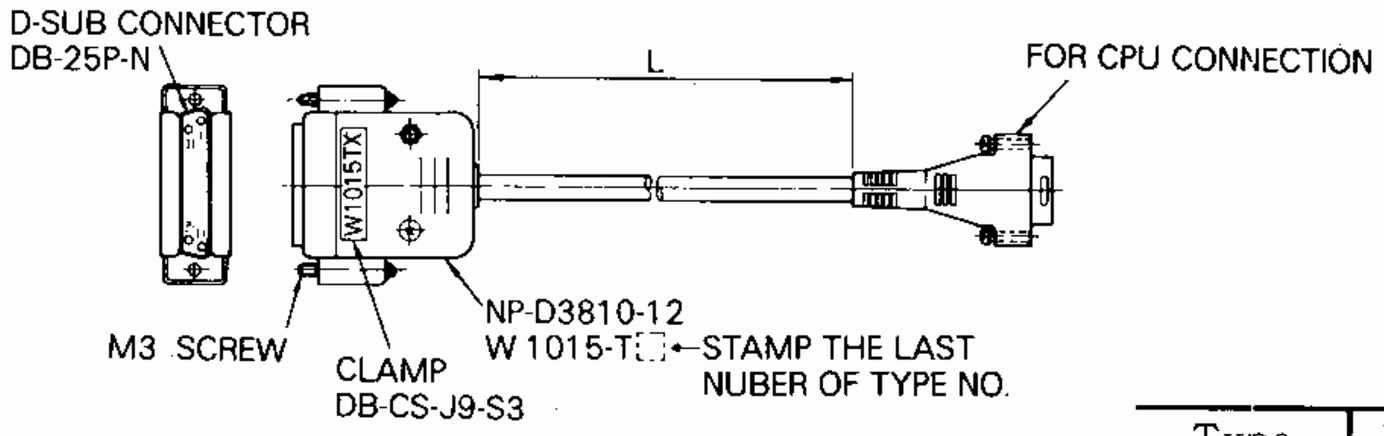
(28) Programming Panel P150 (Type DISCT-P150)



Front View

Side View

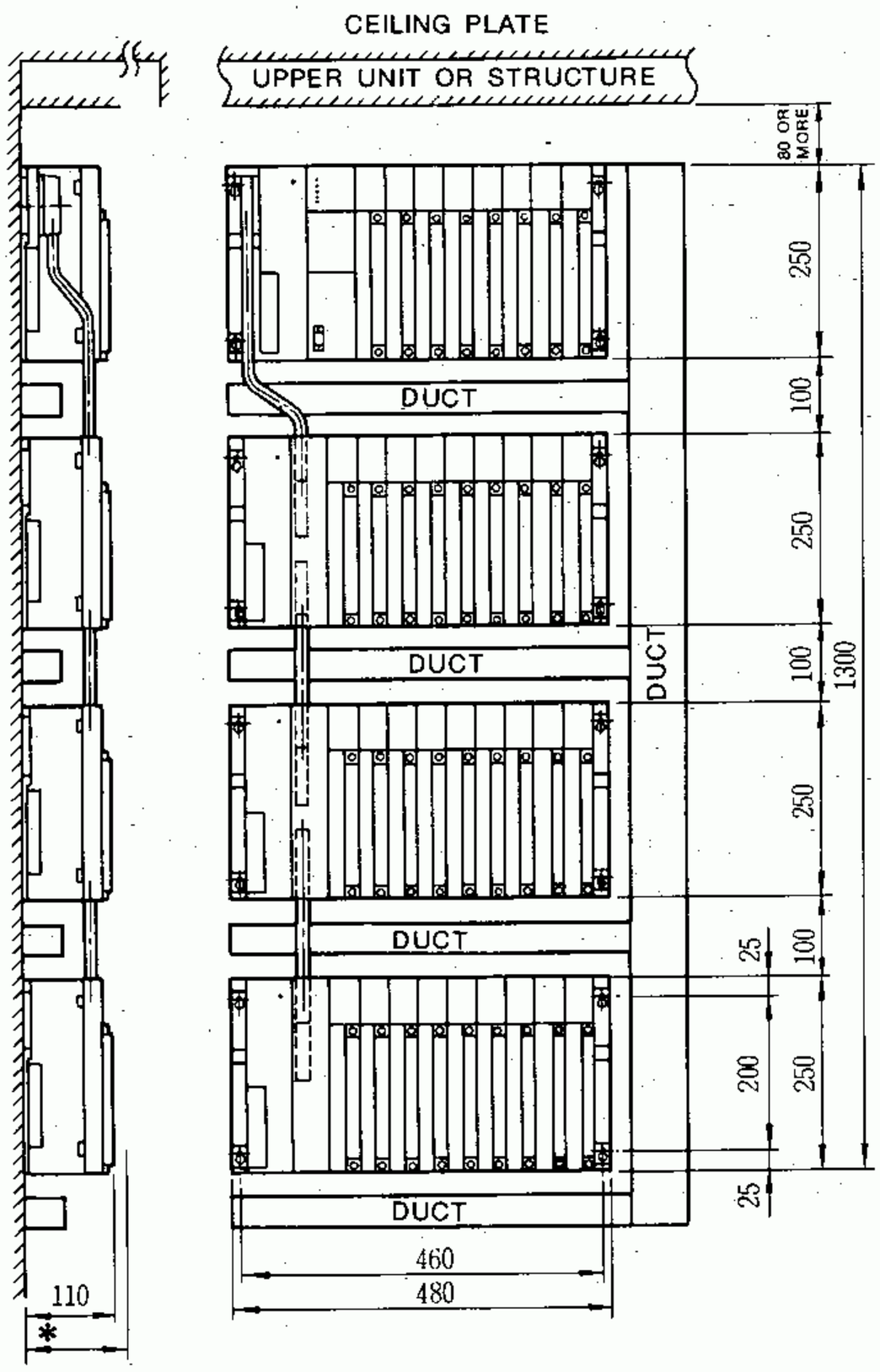
(29) Interface Cable (Type JZMSZ-W1015-T □)

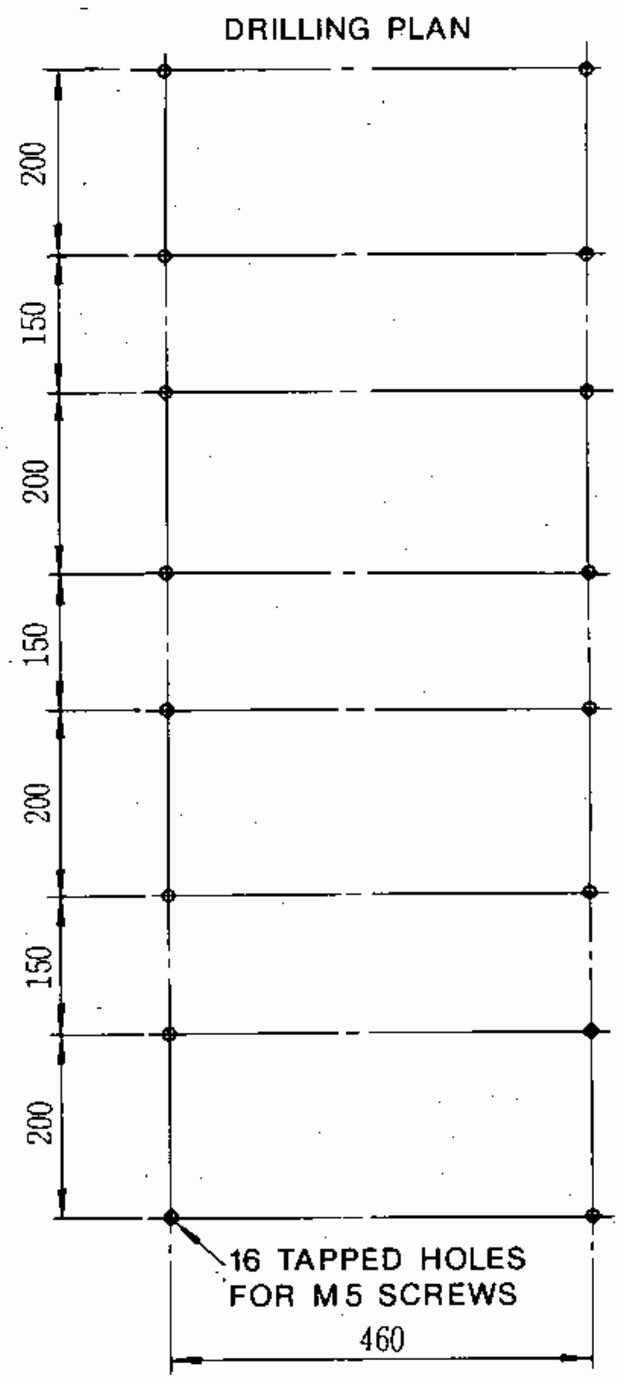


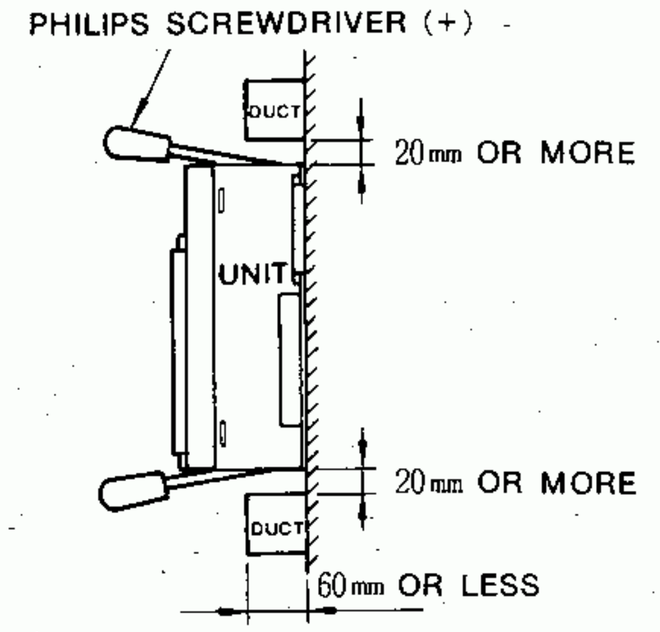
Type	Length	Approx. Weight
W 1015-T 1	2500	0.2kg
W 1015-T 2	15000	1.0kg

APPENDIX C

MEMOCON-SC GL40S LAYOUT AND DRILLING PLAN IN MM







APPENDIX C

MEMOCON-SC GL40S LAYOUT AND DRILLING PLAN IN MM (Cont'd)

MOUNTING PRECAUTIONS

Observe the following when mounting the controller in a frame or other structure. The diagram on the left can be used as a reference.

1. Provide a spacing of more than 80mm from the upper module unit or from the top part of the structure to ensure proper ventilation and for easy module replacement.
2. Apply a philips screwdriver (+) slightly diagonally when mounting or removing a module. Provide spaces at the top and the bottom of modules taking screwdriver and duct sizes into consideration.
3. The mounting side of the mounting base is plated to ensure conduction for better noise resistance. The mounting plate of the frame or of the other structure must also allow conduction with the mounting base.
4. Check the * L dimension (required maximum dimension) in the outline drawings of the modules if the module connectors are mounted on the panel surfaces.

MEMOCON-SC GL40S
DESCRIPTIVE INFORMATION

TOKYO OFFICE
New Pier Takeshiba South Tower, 1-16-1, Kaigan, Minatoku, Tokyo 105 Japan
Phone 81-3-5402-4511 Fax 81-3-5402-4580

YASKAWA ELECTRIC AMERICA, INC.
Chicago-Corporate Headquarters
2942 MacArthur Blvd. Northbrook, IL 60062-2028, U.S.A.
Phone 1-847-291-2340 Fax 1-847-498-2430

Chicago-Technical Center
3160 MacArthur Blvd. Northbrook, IL 60062-1917, U.S.A.
Phone 1-847-291-0411 Fax 1-847-291-1018

MOTOMAN INC. HEADQUARTERS
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277

YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Avenida Brigadeiro Faria Lima 1664-5° CJ 504/511, São Paulo, Brazil
Phone 55-11-815-7723 Fax 55-11-870-3849

YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-888-301

Motoman Robotics AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410

Motoman Robotec GmbH
Kammerfeldstraβe 1, 85391 Allershausen, Germany
Phone 49-8166-900 Fax 49-8166-9039

YASKAWA ELECTRIC UK LTD.
3 Drum Mains Park, Orchardton Woods, Cumbernauld, Scotland, G68 9LD, United Kingdom
Phone 44-1236-735000 Fax 44-1236-458182

YASKAWA ELECTRIC KOREA CORPORATION
Paik Nam Bldg. 901 188-3, 1-Ga Euljiro, Joong-Gu Seoul, Korea
Phone 82-2-776-7844 Fax 82-2-753-2639

YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-282-3003 Fax 65-289-3003

YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677

BEIJING OFFICE
Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-6532-1850 Fax 86-10-6532-1851

SHANGHAI OFFICE
27 Hui He Road Shanghai 200437 China
Phone 86-21-6553-6600 Fax 86-21-6531-4242

YASKAWA JASON (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773

TAIPEI OFFICE
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677

SHANGHAI YASKAWA-TONGJI M & E CO., LTD.
27 Hui He Road Shanghai China 200437
Phone 86-21-6531-4242 Fax 86-21-6553-6060

BEIJING YASKAWA BEIKE AUTOMATION ENGINEERING CO., LTD.
30 Xue Yuan Road, Haidian, Beijing P.R. China Post Code: 100083
Phone 86-10-6233-2782 Fax 86-10-6232-1536

YASKAWA ELECTRIC CORPORATION

YASKAWA

MANUAL NO. SIE-C815-15.1C